Copy No.________

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE DÉFENSE

Direct Linkage of the VAST Finite Element Code to the DSA Database

L. Jiang and M. Norwood
Martec Limited

Martec Limited
1888 Brunswick St., Suite 400
Halifax, NS B3J 3J8

Martec Technical Report: TR-0603

Contract Number: W7707-05-3119/001/HAL

Contract Scientific Authority: D.C. Stredulinsky (902)426-3100 x365

Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2006-185
September 2006

Canada

This page intentionally left blank.

Direct Linkage of the VAST Finite Element Code to the DSA Database

L. Jiang
M. Norwood
Martec Limited

Martec Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia
B3J 3J8

Martec Technical Report: TR-06-30
Contract number: W7707-05-3119/001/HAL
Contract Scientific Authority: D.C. Stredulinsky (902) 426-3100 x365

Defence R&D Canada – Atlantic

Contract Report

DRDC Atlantic CR 2006-185
September 2006

Author

Lei Jiang

Approved by

D.C. Stredulinsky

Contract Scientific Authority

Approved for release by

a. Reid

for ~~Kirk Foster~~

DRP Chair

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Abstract

The general-purpose finite element code VAST, which was developed and maintained by Martec Limited over the past three decades under sponsorship from DRDC Atlantic, has been adopted as the finite element solver engine in several ship structural analysis programs, including the STRUC code developed at DRDC Atlantic. The execution of the VAST program required a set of ASCII input data files and the calculated element stresses were stored in a binary file, named Prefix.V53. In order to assist model generation and stress post-processing, a GUI system, named Trident FEA, has been developed by Martec Limited based on the DSA database. Because a direct linkage between the VAST program and the DSA database was not yet established, all the VAST input data files must be created by Trident or STRUC using information stored in the DSA database and the stress results generated by VAST must be transferred back to the DSA database for plotting. These I/O-intensive, time-consuming operations had to be carried out in an interactive mode, causing a serious efficiency problem for large finite element models with many load cases.

In this contract, a direct linkage of the VAST program to the DSA database was developed. This direct VAST-DSA link provided not only a pre-processing capability for extracting all input information required for VAST executions directly from the DSA database files, but also a post-processing capability to automatically write computed element stresses and element internal forces to the appropriate stress files in DSA database. The most significant advantage of this direct VAST-DSA link was that it integrated the most time-consuming parts of the pre- and post-processing operations into the VAST execution, which was normally conducted in batch mode. This direct link between VAST and the DSA database was verified extensively using a large number of test cases covering all the commonly used element types and analysis options supported by the DSA database. The results are present in this report in detail.

One of the main objectives of the present contract was to speed up the process for generating the LOD file from the load information stored in the DSA database. This was accomplished through modifications in the DSA database routine for extracting load information. Numerical examples indicated that these modifications resulted in a significant reduction on I/O operations, so the speed for LOD file generation was increased by a factor of 5 to 10.

A benchmark on the performance of the direct VAST-DSA link for post-processing was also performed using a practical finite element model. The benchmark results indicated that when multiple load cases existed, the I/O operations on the DSA database files, VES, VEF and VNS, were less efficient than that on the VAST stress file, V53. However, use of the direct VAST-DSA link still has the benefit of having the stress transfer done as a part of the VAST run. To further improve numerical efficiency for populating the DSA stress files, the format of the VES, VEF and VNS may need to be modified.

Other tasks accomplished in this contract included (1) update of the Ansys/DSA translator and (2) enhancement of the HOODFEM program to provide subroutines for extracting mass values and center of gravity co-ordinates for each finite element from the DSA database.

Résumé

Plusieurs programmes de calcul des structures de navires, y compris STRUC, élaboré par RDDC Atlantique, ont adopté, en guise de moteur de résolution par éléments finis, le programme d'analyse par éléments finis à usages multiples VAST qu'a développé et fait évoluer la société Martec Limited au cours des trois dernières décennies, sous le parrainage de RDDC Atlantique. L'exécution du programme VAST nécessitait une série de fichiers de données d'entrée en format ASCII, tandis que les contraintes des éléments calculées étaient stockées dans le fichier binaire Prefix.V53. Pour faciliter la génération des modèles et le post-traitement des contraintes, la société Martec Limited a élaboré l'IUG Trident FEA, en s'appuyant sur la base de données DSA. Puisque aucun lien direct entre VAST et la base de données DSA n'existait encore, tous les fichiers de données d'entrée de VAST devaient être créés au moyen du système Trident ou du programme STRUC, en utilisant des données stockées dans la base de données DSA, et les résultats, sous forme de contraintes, générés par VAST devaient être renvoyés à la base de données DSA pour y être tracés. Ces opérations longues et nécessitant de multiples entrées-sorties devaient être exécutées en mode interactif, ce qui causait un grave problème d'efficacité, dans le cas de modèles d'éléments finis d'envergure, comportant de nombreux cas de charge.

Dans le cadre du présent contrat, on a élaboré un lien direct entre le programme VAST et la base de données DSA. Ce lien direct entre VAST et DSA permet non seulement de se doter d'une capacité de prétraitement en vue d'extraire l'ensemble des renseignements d'entrée nécessaires pour exécuter VAST directement à partir des fichiers de la base de données DSA, mais également d'une capacité de post-traitement, en vertu de laquelle les contraintes et les forces internes des éléments calculées sont directement versées aux fichiers de contrainte appropriés de la base de données DSA. Le principal avantage de ce lien direct entre VAST et DSA tient à l'intégration des parties les plus chronophages des opérations de pré et de post-traitement à l'exécution de VAST, laquelle s'effectue normalement en mode de traitement par lots. On a pu contrôler ce lien direct entre VAST et la base de données DSA de manière approfondie en exécutant un grand nombre de tests couvrant l'ensemble des types d'éléments et d'options en matière d'analyse les plus courants que permet d'envisager la base de données DSA. Les résultats sont présentés de manière détaillée dans le présent rapport.

L'un des principaux objectifs du présent contrat était d'accélérer le processus de génération du fichier LOD à partir des renseignements relatifs aux charges stockés dans la base de données DSA. Pour ce faire, on a apporté des modifications au programme d'extraction des renseignements relatifs aux charges de la base de données DSA. Des exemples numériques ont démontré que ces modifications avaient permis de réduire considérablement les opérations d'entrée-sortie, de telle sorte que le fichier LOG est généré de 5 à 10 fois plus rapidement.

On a également évalué les performances du lien direct entre VAST et DSA sur le plan du post-traitement, en s'appuyant sur un modèle d'éléments finis concret. Les résultats de l'analyse ont démontré qu'en présence de multiples cas de charge, les opérations d'entrée-sortie visant les fichiers de la base de données DSA, VES, VEF et VNS étaient moins efficaces que celles qui visaient le fichier des contraintes VAST, V53. Cependant, le recours au lien direct entre VAST et DSA présente l'avantage d'intégrer le transfert des contraintes au passage VAST en tant que tel. Il pourrait s'avérer nécessaire de modifier le format des fichiers

VES, VEF et VNS pour améliorer encore l’efficacité numérique de la constitution des fichiers des contraintes DSA.

Parmi les autres tâches réalisées dans le cadre du présent contrat figuraient (1) une mise à jour du programme de traduction Ansys/DSA et (2) une amélioration du programme HOODFEM en vue d’élaborer des sous-programmes permettant d’extraire les valeurs des masses ainsi que les coordonnées des centres de gravité de chacun des éléments finis de la base de données DSA.

Executive summary

Introduction: The general-purpose finite element code VAST, which was developed and maintained by Martec Limited over the past three decades under sponsorship from DRDC Atlantic, has been adopted as the finite element solver engine in several ship structural analysis programs, including the STRUC code developed at DRDC Atlantic. The STRUC code has been developed through the support of the Cooperative Research Ships (CRS) organization headed MARIN (Maritime Research Institute Netherlands) for prediction of long term structural response for fatigue and strength assessment. The execution of the VAST program required a set of ASCII input data files and the calculated element stresses were stored in a binary file, named Prefix.V53. In order to assist model generation and stress post-processing, a GUI system, named Trident FEA, has been developed by Martec Limited based on DSA database. Because a direct linkage between the VAST program and the DSA database was not yet established, all the VAST input data files must be created by Trident or STRUC using information stored in the DSA database and the stress results generated by VAST must be transferred back to the DSA database for plotting. These I/O-intensive, time-consuming operations had to be carried out in an interactive mode, causing a serious efficiency problem for large finite element models with many load cases.

Principal Results: A direct linkage of the VAST program to the DSA database was developed. This includes a pre-processing capability for extracting all input information required for VAST executions directly from the DSA database files, and also a post-processing capability to automatically write computed element stresses and element internal forces to the appropriate stress files in DSA database.

Significance of Results: With typical large full-ship structural finite element models used in STRUC, VAST generates very large binary files containing stress results for the large number of load cases involved in this process. Since STRUC uses the DSA database to store finite element model data and results, the stress results must be read from the VAST binary output files and stored in the DSA database before they can be processed in STRUC. This is a time consuming process that has now been eliminated by creating an option in VAST to write stress results directly into the DSA database. The new process of populating the DSA stress files directly from VAST was less efficient than population of the original VAST V53 stress file, but the overall process is significantly faster and the DSA database stress files were typically a factor of two smaller than the original VAST binary stress files.

Future Plans: Additional investigation may be conducted to determine if the numerical efficiency for populating the DSA stress files may be improved through modification of the format of the VES, VEF and VNS database files.

Jiang L., Norwood M. 2006. *Direct Linkage of the VAST Finite Element Code to the DSA Database*. DRDC Atlantic CR 2006-185. Defence R&D Canada - Atlantic.

Sommaire

Introduction : Plusieurs programmes de calcul des structures de navires, y compris STRUC, élaboré par RDDC Atlantique, ont adopté, en guise de moteur de résolution par éléments finis, le programme d'analyse par éléments finis à usages multiples VAST qu'a développé et fait évoluer la société Martec Limited au cours des trois dernières décennies, sous le parrainage de RDDC Atlantique. Le programme STRUC a été élaboré avec l'appui de l'organisation Cooperative Research Ships (CRS) dirigée par le Maritime Research Institute Netherlands (MARIN) en vue de prévoir la réaction structurale à long terme, en matière de fatigue et de résistance aux sollicitations externes. L'exécution du programme VAST nécessitait une série de fichiers de données d'entrée en format ASCII, tandis que les contraintes des éléments calculées étaient stockées dans le fichier binaire Prefix.V53. Pour faciliter la génération des modèles et le post-traitement des contraintes, la société Martec Limited a élaboré l'IUG Trident FEA, en s'appuyant sur la base de données DSA. Puisqu'aucun lien direct entre VAST et la base de données DSA n'existait encore, tous les fichiers de données d'entrée de VAST devaient être créés au moyen du système Trident ou du programme STRUC, en utilisant sur des données stockées dans la base de données DSA, et que les résultats, sous forme de contraintes, générés par VAST devaient être renvoyés à la base de données DSA, pour y être tracés. Ces opérations longues et nécessitant de multiples entrées-sorties devaient être exécutées en mode interactif, ce qui causait un grave problème d'efficacité, dans le cas de modèles d'éléments finis d'envergure, comportant de nombreux cas de charge.

Résultats principaux : On a élaboré un lien direct entre le programme VAST et la base de données DSA. Ce lien permet non seulement de se doter d'une capacité de prétraitement en vue d'extraire l'ensemble des renseignements d'entrée nécessaires pour exécuter VAST directement à partir des fichiers de la base de données DSA, mais également d'une capacité de post-traitement, en vertu de laquelle les contraintes et les forces internes des éléments calculées sont directement versées aux fichiers de contrainte appropriés de la base de données DSA.

Importance des résultats : Dans le cas des modèles, généralement imposants, de calcul des structures par éléments finis d'un navire complet auxquels on a recours avec le programme STRUC, VAST génère de très gros fichiers binaires de résultats, sous forme de contraintes, pour la multiplicité des cas de charge prévus en vertu d'un tel processus. Puisque STRUC a recours à la base de données DSA pour stocker les données et les résultats du modèle d'analyse par éléments finis, les résultats, sous forme de contraintes, doivent être lus à partir des fichiers de sortie binaires VAST et stockés dans la base de données DSA avant qu'ils puissent être traités au moyen de STRUC. Il s'agit d'un long processus que la création d'une option, intégrée à VAST, en vertu de laquelle il est désormais possible de consigner directement les contraintes dans la base de données DSA, a éliminé. Si le nouveau processus de constitution des fichiers de contraintes DSA directement à partir de VAST s'avère moins efficace que celui qui consistait à constituer le fichier des contraintes V53 VAST d'origine, le processus, dans son ensemble, est considérablement plus rapide et les fichiers des contraintes de la base de données DSA sont généralement deux fois plus petits que les fichiers des contraintes binaires VAST d'origine.

Travaux futurs : D’autres analyses pourraient être menées afin de déterminer s’il serait possible d’améliorer l’efficacité numérique de constitution des fichiers des contraintes DSA en modifiant le format des fichiers de la base de données VES, VEF et VNS.

Jiang L., Norwood M. 2006. *Direct Linkage of the VAST Finite Element Code to the DSA Database*. RDDC Atlantique CR 2006-185. R et D pour la défense Canada - Atlantique.

Table of contents

List of figures

List of tables

This page intentionally left blank.

1. Introduction

The general-purpose finite element code VAST, developed and maintained by Martec Limited over the past three decades under sponsorship from DRDC Atlantic, has been adopted as the finite element solver engine in several ship structural analysis programs, such as the IST for predicting crack initiation and propagation on the CPF, and SubSAS for analysis and design of submarine structures. Similar to many other finite element programs, VAST relied on a number of ASCII files to define all input information required for finite element analyses, including nodal co-ordinates and element connectivities, material properties, lumped masses, fluid added masses, loading conditions and boundary conditions in finite element models. Upon completion of an analysis, VAST generated an ASCII result file (LPT) and several binary files, which stored results of eigenvalue analyses (V51), displacements (V52) and element stresses (V53). These binary files were subsequently used for post-processing.

A pre- and post-processing program, named Trident FEA, was also developed by Martec Limited to generate input data files for the VAST finite element solver engine and to post-process the finite element results generated by VAST. The Trident FEA program utilized a database, called DSA, to store all information required to define a finite element model, and the calculated element stresses and strains in all elements for all load cases, eigen modes, solution steps and time steps, depending on the type of analyses. Because the VAST solver had no direct linkage to the DSA database, the user had to generate the ASCII input data files before a VAST execution, and transfer the element stress/strain results back to the DSA database after a VAST analysis. This data handling procedure was also adopted in the HOOD-based STRUC code, in which the DSA database was accessed in the way through HOODFEM to facilitate data communication with the VAST program. However, this procedure could be very time-consuming, especially when it was used in a GUI program that must be executed in an interactive mode. In addition, this procedure also led to excessive use of disk space due to duplication of information.

The objective of the present contract was to remove these difficulties by developing a direct linkage of the VAST finite element program to the DSA database. Through this direct VAST-DSA link, VAST could extract information related to a finite element model directly from the DSA database, and write the analysis results directly to the DSA database files. This modification of the VAST program not only improved the overall efficiency for performing finite element analyses using VAST, but also integrated the I/O operations to the DSA database into the VAST execution, which could be carried out in batch mode for large problems.

In this report, details on the development of the direct linkage between VAST and DSA database is presented in the next chapter for both post- and pre-processing. The control parameters for this direct linkage are provided at the beginning of the USE file using a keyword-based approach. A detailed instruction for defining these control parameters in the Trident FEA program is presented. In order to verify this newly developed VAST-DSA link, extensive numerical experiments were performed which covered all commonly used element types and analysis options supported by the DSA database. A detailed presentation of the test results is provided in Chapter 3. Also provided in Chapter 3 are some benchmark results on the relative numerical performance of the new VAST-DSA link against the original method of data transfer between these two programs. Concluding remarks are given in Chapter 4.

2. Development of a Direct VAST-DSA Linkage

2.1 Post-Processing

The development of post-processing capabilities in the direct VAST-DSA linkage required provision of options to write various finite element analysis results directly into DSA database files. These included the element stresses or strains to the VES file, element internal forces to the VEF file and element nodal stresses to the VNS file. A list of all the element types and analysis options supported by the current version of DSA database is given in Table 2.1. A summery of supported nonlinear analysis options is presented in Table 2.2, where GN, EP and HE indicate geometric nonlinearity, elasto-plasticity and hyper-elasticity, respectively.

In order to establish the direct link with the DSA database for post-processing, various DSA interface routines were implemented in the STRESS module of the VAST program to handle the database files and write element and nodal stress/strain values and nodal forces to the appropriate files in a format required by DSA. The same subroutines that were used to process stress data in Trident FEA were utilized directly in VAST. This arrangement not only avoided redundant code, but also improved maintainability of the program.

As required in the Statement of Work, the modified VAST program should provide options to directly write element stresses and internal forces to the VES, VNS and VEF files in the DSA database during a VAST execution, but also maintain the original binary stress/strain file, V53. To satisfy the requirements for generating different stress files, a new control flag, named IT53, was introduced in the STRESS module, which provided the following three options:

IT53	=	0, element stresses (or strains) will be stored on the V53 file.
	>	0, element stresses (or strains) will be stored on the DSA database files.
	<	0, element stresses (or strains) will be stored on both the V53 file and the DSA database files.

In a VAST execution, the direct VAST-DSA link is activated by a set of keyword-driven input commands at the beginning of the Prefix.USE file. These keywords and their associated flags are summarized in Table 2.3. Among them, two keyword commands, *db_option and *db_results, are related to the transfer of element stresses/strains and internal forces to the DSA database files, Prefix.VES, Prefix.VNS and Prefix.VEF. The first keyword specifies whether element results are to be transferred to the DSA database, whereas the second keyword permits the user to define details for this stress transfer. These details include decisions on average or maximum stresses, stresses or strains, layer number for composite shells, and whether the element nodal stresses and element internal forces are to be transferred.

For linear static and dynamic analyses and eigenvalue analyses, VAST provides an option for calculating either stresses or strains, through a flag named IOPT under header ISTRES. In these cases, the option indicated by the *db_results command must be consistent with the value of flag IOPT. For nonlinear static and dynamic analyses, IOPT should always equal to 1 as element stresses are required to formulate internal force vector in equilibrium iterations.

However, for analyses involving elasto-plastic materials, the user can choose to output plastic strains to the DSA database by specifying element strains in the *db_results command. For all analysis types, the specification for element internal forces must be consistent with the value of flag IELMFOR in the ISTRES section.

For highly nonlinear problems, it was normally impossible to ensure that the first analysis would result in a complete solution. When divergence occurred, the users would have to make some adjustments to the solution control parameters and restart the nonlinear run from the point where numerical problems started to develop. Because this restarting point might be in the middle of the previous run, the previous generated stress files must be truncated at the proper location. Another issue for nonlinear analysis was related to the reduction of stress files. Because nonlinear analyses normally involved a large number of solution steps and required stresses and other parameters related to the constitutive relations to be stored at each numerical integration point, the size of stress files could be very large. To reduce the size of stress files, VAST permitted the user to define a step interval for permanent storage of the stress results. In this case, restart would only be possible from the solution steps at which the stress data were available. In the present development of the direct VAST-DSA linkage, these features for treating stress files were extended to DSA database files, VES, VNS and VEF. However, because these DSA database files did not contain stress information at each numerical integration point, they were insufficient to restart a nonlinear analysis. For this reason, if a restart is anticipated, the Prefix.V53 must be generated by setting IT53<0 or =0.

In order to assist creation of these keyword-driven commands in the Prefix.USE file, new dialog boxes were generated in the Trident GUI to allow the user to indicate whether the direct VAST-DSA linkage is going to be utilized and in case of Yes, to generate the proper USE file based on user's specifications. An instruction on how to use these new dialog boxes, as well as sample Prefix.USE files involving the direct VAST-DSA linkage, are presented later in this Chapter.

2.2 Pre-Processing

In addition to the post-processing capabilities in the direct DSA-VAST link as described above, pre-processing capabilities were also developed in the direct VAST-DSA link to automatically extract input information required by VAST executions. In the present work, subroutines from the DSA database were incorporated directly into the VAST code to access the DSA database and generate a set of dummy input data files required by VAST. These files include:

- GOM for nodal coordinates and element information;
- SMD for boundary conditions, multipoint constraints and rigid links;
- MMD for lump masses;
- AMD for fluid added masses;
- DIS for prescribed displacements; and
- LOD for external element and concentrated loads.

The main advantage for taking this approach was that it avoided extensive modifications and verifications of the existing modules in the VAST program. The disadvantage of this approach was computational cost associated with the generation of the ASCII data files.
However, our experience indicated that this extra step only increased the total run time of the VAST analyses marginally for practical sized finite element models. This small increase of run time should not cause any significant problem especially when VAST is executed in the batch mode.

As mentioned in the Statement of Work for the present contract, one of the main purposes for the development of a direct VST-DSA linkage was to speed up the process of generating the LOD file from the load information in DSA database. The past experience with Trident FEA indicated that for a large finite element model with a large number of load cases, the generation of the LOD file alone from the Trident GUI could take over one hour. An investigation of this problem indicated that this inefficiency was associated with an inconsistency between the orders of the load information stored in the DSA database and in the ASCII LOD file. When multiple load cases existed, the element loads on an element for all load cases were stored together on a single record in the DSA database. However, the ASCII LOD file was organized in such a way that all the element and concentrated loads in a load case were grouped together. As a result, when the ASCII LOD file was created, the DSA database file must be scanned through repeatedly for each load case, which caused significant I/O operations.

In the present contract, the DSA database routine for creating the ASCII LOD file was modified for improved efficiency. In the modified program, two new integer arrays were introduced to hold the element and node numbers on which external loads were applied in each load case. Making use of these arrays, the I/O operations required during generation of the ASCII LOD file was considerably reduced. A benchmark test using a global CPF model with 207 load cases indicated that these modifications increased the speed of LOD file generated by a factor of 5.

Similar to the post-processing capabilities described before, the pre-processing capabilities in the direct VAST-DSA linkage were also controlled by keyword-driven commands listed in Table 2.3. In these commands, the need for generating the dummy input data files was indicated by keyword *db_option. The generations of the LOD and DIS files were controlled by the input parameter associated keyword *db_analysistype, which defined the type of desired analysis. The creations of the boundary condition file (SMD), the lumped mass file (MMD) and the fluid added mass file (AMD) were also controlled by keywords listed in Table 2.3.

2.3 Generation of USE File for Using the Direct VAST-DSA Linkage

To demonstrate the usage of the direct VAST-DSA linkage, we considered linear static analysis of a simply supported plate subjected to a uniform pressure as shown in Figure 2.1. In the updated Trident GUI, an additional item was provided in the dialog box for Analysis Options, as shown in Figure 2.2, to allow the user to indicate whether the DSA database was going to be accessed during the present VAST execution. If the answer was yes, another dialog box, as shown in Figure 2.3, would appear, permitting the user to enter several control parameters related to the DSA database. The Prefix.USE file created by the updated Trident GUI is displayed in Figure 2.4, where a section of keyword-driven commands defining operations on the DSA database files were inserted at the beginning of the USE file. In the modified Trident FEA code, all the DSA database files were closed before the VAST execution to allow these files be opened and modified by VAST during analysis. Once the VAST execution is completed, the DSA database files were reopened in Trident for plotting the stress results.

For displaying stress results, the first item in the dialog box shown in Figure 2.5, namely Transfer to Database, was no longer necessary since the DSA database files were already generated by VAST and were ready for being used to create stress plots. A typical stress contour plot, showing distribution of the von Mises stress on the top surface of the plate, is given in Figure 2.6. A contour plot for the same stress distribution was also generated using the old procedure, which required

transfer of stress data from the V53 file to the DSA database in Trident GUI. This later stress contour plot was found to be identical to Figure 2.6.

An extensive numerical verification of the direct VAST-DSA link, which covered all commonly used element types and analysis options, is presented in the next Chapter.

Table 2.1: Element Types and Analysis Options Supported by the DSA Database.

IEC	Description	Data Types & Analysis Options				
		Element Stresses/ Strains	Nodal Stresses/ Strains	Element Forces	Composite	Nonlinear
1	8-noded shell	√	√	√	√	√
2,52	20-noded solid	√	√	√		√
3	2-noded beam	√		√		√
4	3-noded plate	√		√	√	√
5	4-noded shell	√	√	√	√	√
6	13-noded transition	√		√		
7	3-noded beam	√		√		√
8	2-noded bar	√		√		
9	3-noded membrane	√		√		
10	3-noded bar	√		√		
11	4-noded membrane	√		√		
16,66	8-noded solid	√	√	√		√
17	10-noded solid	√		√		
18	4-noded fracture	√		√		
19	3-noded axisym shell	√				
20	8-noded membrane	√		√		
21	8-noded fracture	√		√		
22	1-noded axisym beam	√				
23	4-noded shear web	√		√		
24	20-noded fracture	√		√		
55	4-noded stiffened shell	√	√	√	√	

Table 2.2: Nonlinear Analysis Options Supported by the DSA Database.

IEC	Description	Nonlinear Material Property		
		GN	EP	HE
1	8-noded thick/thin shell	√	√	
2,52	20-noded solid	√	√	√
3	2-noded general beam	√	√	
4	3-noded triangular plate	√	√	
5	4-noded quad shell	√	√	
7	3-noded beam	√	√	
8	2-noded bar/cable	√	√	
16,66	8-noded solid	√	√	√

Table 2.3: Keywords and Control Parameters for Direct VAST-DSA Linkage.

Parameter	Value	Description
*db_option= I1		
I1	1	Model generation and results transfer
	2	Model generation only
	3	Results transfer only
*db_filename= C1		
C1		Prefix of the DSA database
*db_analysis= I1		
I1	10	static, no loads (not active)
	11	static, element/concentrated loads
	12	static, support motion
	13	static, element/concentrated loads plus prescribed displacements
	14	prescribed displacements only
	20	dynamic, direct integration, no loads
	21	dynamic, direct integration, element and concentrated loads
	22	dynamic, direct integration, support motion
	23	dynamic, modal superposition, element and concentrated loads
	24	dynamic, modal superposition, support motion
	25	dynamic, modal frequency response, element and concentrated loads
	26	dynamic, modal frequency response, support motion
	27	dynamic, direct frequency response, element and concentrated loads
	28	dynamic, direct frequency response, support motion
	29	dynamic, response spectrum
	31	eigenvalue, direct iteration
	32	eigenvalue, subspace iteration
	40	complex frequency
	51	temperature loading, no loads
	52	temperature loading, element and concentrated loads
*db_results= I1 I2 I3 I4 I5		
I1	1	Average element stresses/strains are written to DSA database
	2	Maximum element stresses/strains are written to DSA database
I2	1	Element stresses are written to DSA database
	2	Element strains (for linear analysis) or plastic strains (for nonlinear analysis involving elastic-plastic material) are written DSA database
I3	0	Element nodal stresses are not written to DSA database file, VNS.
	1	Element nodal stresses are written to DSA database file, VNS.
I4	0	Element nodal forces are not written to DSA database file, VEF.
	1	Element nodal forces are written to DSA database file, VEF.
I5	>0	Layer number for which stresses/strains are written to DSA database
	-1	Stresses/strains on top and bottom surfaces of the entire composite plate are written to DSA database

*db_smd		
		Boundary conditions are involved in the present analysis
*db_amd		
		Fluid added masses are involved in the present analysis
*db_mmd		
		Lumped masses are involve din the present analysis
*db_enddata		
		End of input data section for direct VAST-DSA link

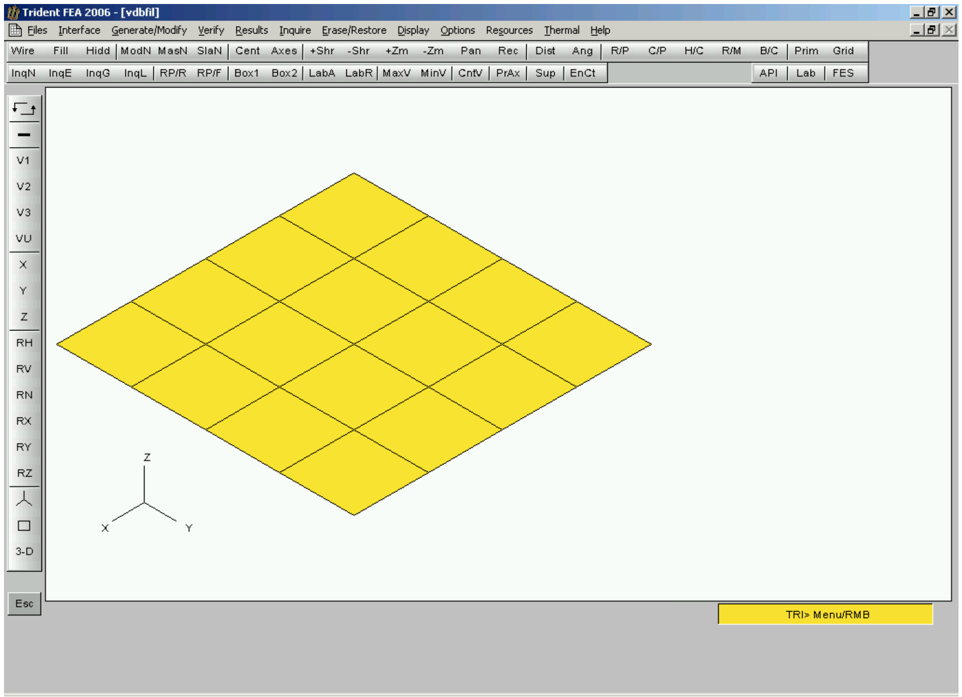


Figure 2.1: Finite Element Model for Linear Static Analysis of a Simply Supported Plate Subjected to Uniform Pressure.

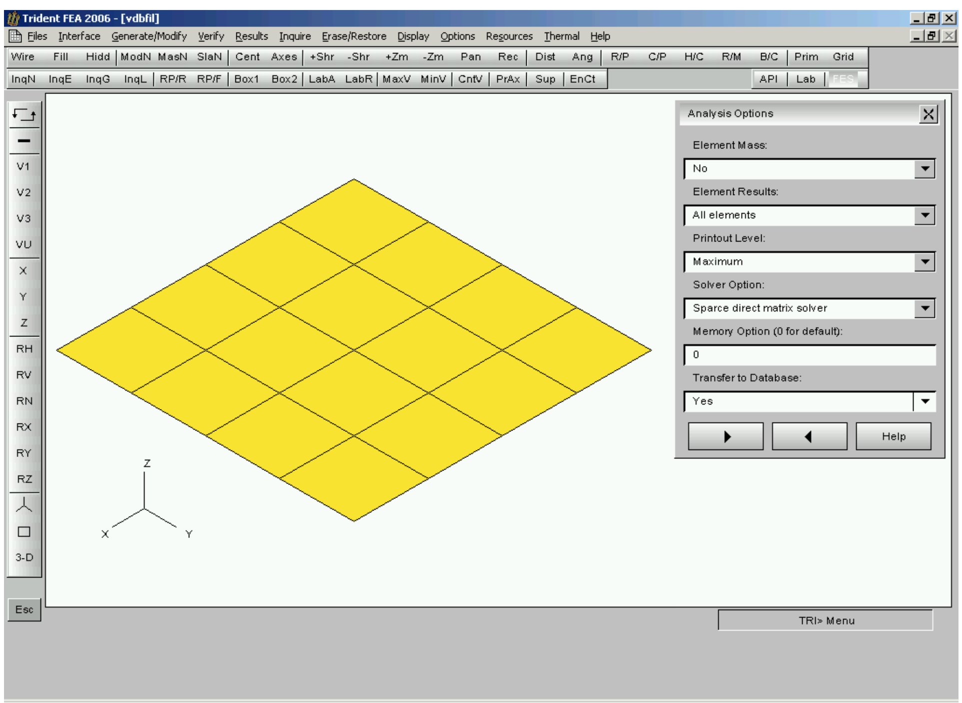


Figure 2.2: Dialog Box for Defining Analysis Options.

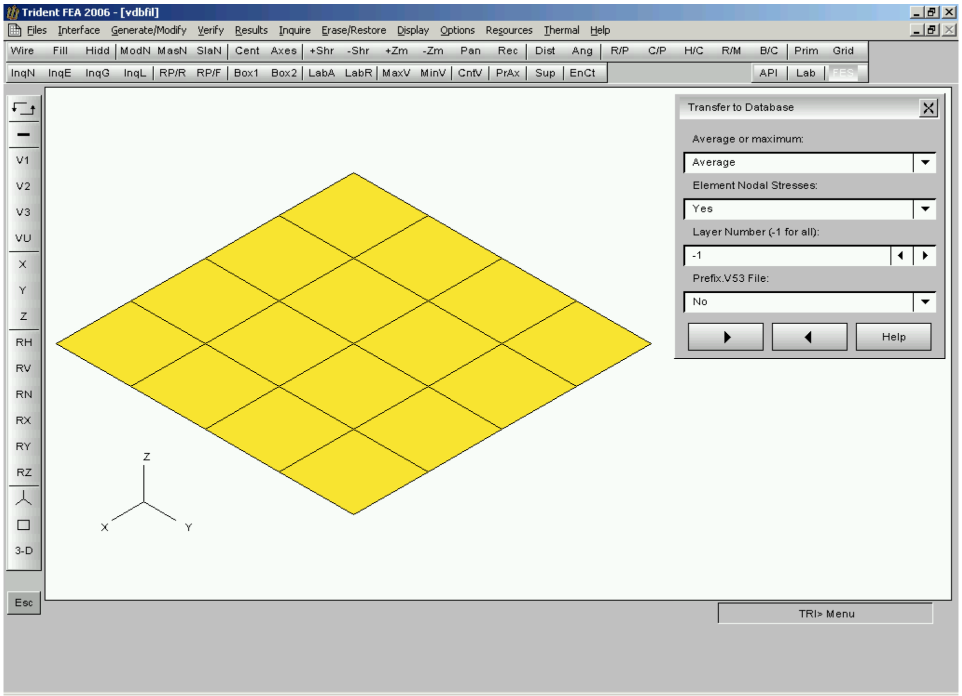
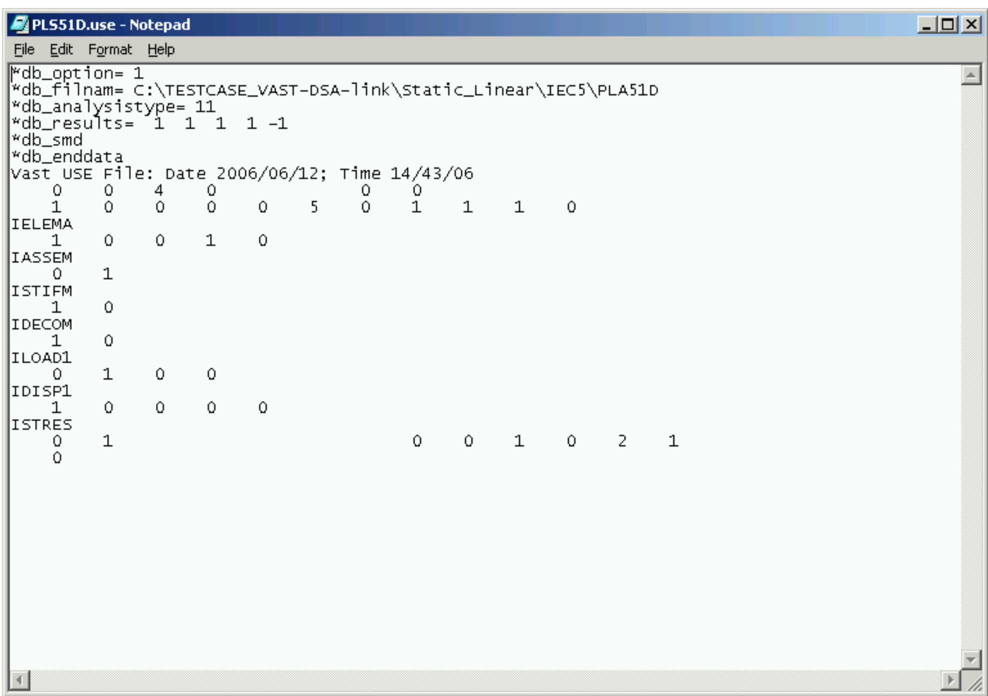


Figure 2.3: Dialog Box for Defining Control Parameters for Transferring Stress Results to the DSA Database Files.



```
*db_option= 1
*db_filnam= C:\TESTCASE_VAST-DSA-link\Static_Linear\IEC5\PLA51D
*db_analysistype= 11
*db_results=  1  1  1  1 -1
*db_smd
*db_enddata
Vast USE File: Date 2006/06/12; Time 14/43/06
     0     0     4     0                 0     0
     1     0     0     0     0     5     0     1     1     1     0
IELEMA
     1     0     0     1     0
IASSEM
     0     1
ISTIFM
     1     0
IDECOM
     1     0
ILOAD1
     0     1     0     0
IDISP1
     1     0     0     0     0
ISTRES
     0     1                             0     0     1     0     2     1
     0
```

Figure 2.4: Prefix.USE File Containing Keyword-Driven Commands Defining Operations Using the Direct VAST-DSA Link.

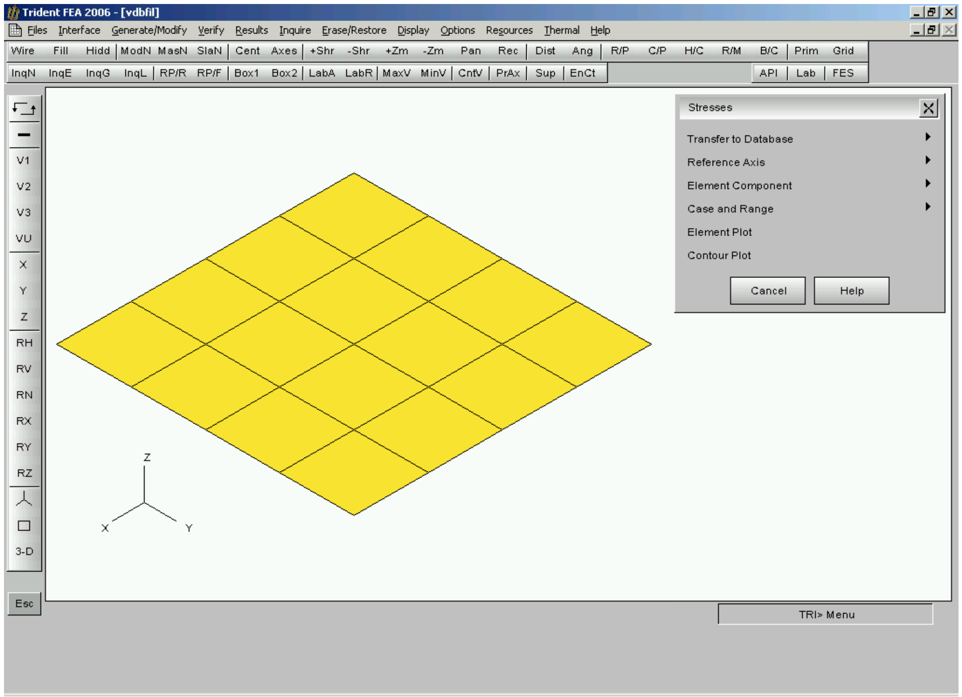


Figure 2.5: Dialog Box for Generating Stress Plots.

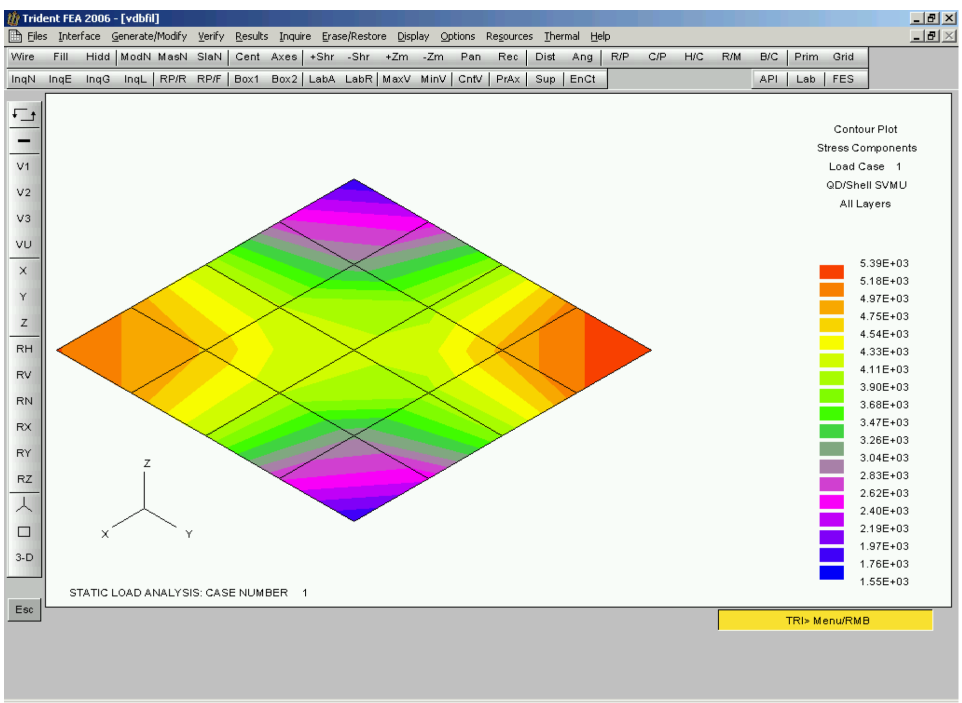


Figure 2.6: A Typical Stress Contour Plot Generated Using the VES File Populated Through the Direct VAST-DSA Link.

3. Verification and Benchmarking of the Direct VAST-DSA Linkage

3.1 Numerical Verification

The newly developed direct linkage in the VAST program to the DSA database was verified extensively using a large number of example problems involving all the commonly used element types and analysis options supported by the DSA database. These test problems involved a wide range of complexity, from a simple plate to the entire CPF model. It should be noted that the purpose of this numerical verification was not to demonstrate the computational capabilities of the Trident/VAST finite element system, but to check the correctness of the direct VAST-DSA link developed in the present contract. For this reason, the test problems were selected to be as simple as possible, but also sufficient for verifying the pre- and post-processing features in the VAST-DSA link.

In this numerical verification process, the DSA database files containing input information for finite element analyses, such as nodal co-ordinates, element connectivity, material properties and boundary conditions, were first populated using exiting ASCII input files for the VAST code. The Prefix.USE file was then generated using the “Analysis Options” dialog box provided in the Trident program, in which the VAST-DSA link was selected. During the VAST execution, a set of dummy ASCII input files, i.e. GOM, SMD, LOD *et al*, were generated. In the present version of VAST, these intermediate ASCII data files were maintained at the end of the VAST runs, so that they could be verified with comparison with the original ASCII files used to establish the DSA database.

To verify the pre-processing capability, a wide range of solution options for linear and nonlinear, static, dynamic and modal analyses were considered, including:

- Linear elastic material
- Composite material
- Geometric nonlinearities
- Elastic-plastic and hyper-elastic materials
- Multi-point constraints
- Multiple load cases
- Modal stresses
- Dynamic response through direct time integration

For post-processing, the stresses and strains plots generated by Trident FEA were obtained using two different methods. One was to use the DSA database files directly generated during VAST runs through the direct VAST-DSA link, and the other was to use the Trident capability to transfer stresses from the Prefix.V53 file to the DSA database files. The stress and strain plots obtained using different methods were compared with each other and also compared with the stress and strain values in the LPT file. Through these exercises, a few minor errors were identified in the DSA database routines for dealing with element stresses. These errors were corrected.

All the verification problems developed in the present contract were delivered as a part of the package of final deliverables to the Scientific Authority. These test models were arranged in a directory

system named by the analysis types. Within each directory, test cases for different element types were stored in different sub-directories named by the elements' IEC number. Each sub-directory contained a complete set of the original ASCII input data files, an archived DSA database file, a Prefix.USE file using the direct VAST-DSA link, and a LPT file generated by VAST through the direct VAST-DSA link. A brief description of these test cases and results are provided in the following paragraphs in this chapter. Because detailed information regarding the test cases, such as dimensions and material properties, were available in the ASCII data files, they were not repeated in this report.

3.1.1 Linear Static Analysis

In order to verify the direct VAST-DSA linkage for linear static analysis using shell and 3D solid elements, we considered a simply supported square plate subjected to uniform pressure loads. Due to symmetry in this problem, only one quarter of the plate was included in the finite element model. To test the capability of the direct VAST-DSA link for treating multiple load cases, three load cases, which involved different magnitudes of uniform pressure, were considered.

Five element types, including 8-noded shell (IEC1), 20-noded solid (IEC2), 3-noded triangular plate (IEC4), 4-noded quad shell (IEC5), 4-noded stiffened shell (IEC55) and 8-noded solid (IEC16), were utilized to idealize the simply supported plate and the results generated by Trident FEA were presented in Figures 3.1-3.12. These figures included element stress plots for user selected stress component, and data tables containing element nodal stresses and element nodal forces in user specified element. It was worth mentioning that the VAST code provided an option to only calculate stresses in user-selected elements. This feature was also maintained in the direct VAST-DSA link. Figure 3.8 contained an element stress plot where stresses were only generated in four elements. In order to make Trident FEA generate this plot properly, special treatments were required to the VES file.

Figures 3.1-3.12 were generated using DSA database files created using the direct VAST-DSA linkage. As mentioned before, these plots were compared with those obtained through stress transfer from the V53 to the DSA database. The stress values reported in these plots were also compared with the stress solutions in the LPT files. Results generated from all three sources were found to be identical.

The linear static capability of beam elements was verified using a cantilever beam subjected to a concentrated force at its free end. The beam, which had a solid rectangular cross-section, was represented by 2-noded (IEC3) and 3-noded (IEC7) beam elements. Element stress plots were generated using different methods described above for points on top and bottom surfaces of beam cross-section. Exact agreement between these plots was identified. Due to space limitation, these plots were not included in the report.

The other test examples for linear static analysis include:

- Square membrane subjected to uniform tension (IEC9, IEC11, IEC20)
- Stiffened cylinder subjected to uniform pressure load (IEC19, IEC22)
- Plate with an edge crack (IEC18, IEC21)

To verify the pre- and post-processing capabilities of the direct VAST-DSA link for multi-layer composite shells, we still considered a simply supported plate subjected to uniform pressure. However, differing from the previous test case, the plate in the current example contained four layers

of orthotropic materials with a ply sequence [0/90/0/90] from bottom to top. All four of the composite shell elements supported by the DSA database, namely 8-noded shell (IEC1), 3-noded triangular plate (IEC4), 4-noded quad plate (IEC5) and 4-noded stiffened plate (IEC55), were employed in this test example. When the direct VAST-DSA link was selected, all the input data were directly retrieved from the DSA database. The only exception was the Prefix.CMP file, which contained material properties of each laminar in the composite shell.

For post-processing, the user had the freedom to specify the layer for which stresses are to be written to the DSA database. Figures 3.13 and 3.14 show the element stress plots for the upper and lower surfaces of the first layer, which is the bottom layer in the composite shell. Taking the same approach, element stress plots for the top layer were also obtained and displayed in the Figures 3.15 and 3.16. When a “-1” was entered for the layer number, stresses on the very top and bottom surfaces of the entire composite plate were transferred to DSA database, and the corresponding element stress plots are presented in Figures 3.17 and 3.18.

It is easy to notice that all the verification examples described above involved a single element type. However, in practical finite element analyses, multiple element types are often utilized to represent a geometrically complicated structure. To verify the direct VAST-DSA link for models involving multiple element types, we considered a global finite element model of a CPF with 207 load cases. The finite element mesh and summaries of element types and load cases are given in Figures 3.19 and 3.20. The predicted displacement and stress contours for the final load case were given in Figures 3.21 and 3.22, respectively. The element nodal stresses and forces in an element in the main deck for the last load case are shown in Figures 3.23 and 3.24. This example problem was also used to benchmark the efficiency of the direct VAST-DSA link. The results will be discussed later in this Chapter.

3.1.2 Nonlinear Static Analysis

For nonlinear static analyses, we first considered geometrically nonlinear response of a simply supported plate under uniform pressure. Five element types were utilized to solve this problem, including 8-noded shell (IEC1), 3-noded triangular plate (IEC4), 4-noded quad shell (IEC5), 20-noded solid (IEC2) and 8-noded solid (IEC16) elements. The undeformed and deformed meshes of 8-noded shell element are given in Figure 3.25 and the load-center deflection curve predicted by the 8-noded shell elements was presented in Figure 3.26. This nonlinear curve indicated a stress-hardening behaviour, which is typical to this type of nonlinear problems. The element stress plots and the tables for element nodal stresses and forces generated through the direct VAST-DSA link were presented in Figures 3.27-3.39.

These figures were compared with those obtained by transferring stresses from the V53 file to the DSA database and the stress values printed into the LPT file. Excellent agreement between all three sources is observed.

For geometric nonlinear analysis of beam structures, we considered large displacement and large rotation of a cantilever beam subjected to an end point load. Two finite element models were utilized in which the cantilever beam was discretized into six 2-noded beam elements and three 3-noded beam elements, respectively. The original and deformed beam configurations predicted by the 2-noded elements were presented in Figure 3.40, whereas the element stress plot at a point on the top surface of beam cross-section and a table showing element force are given in Figures 3.41 and 4.42. The final axial stress distributions at points on the top and bottom surfaces of the cross-section predicted

by the 3-noded beam elements are presented in Figures 3.43 and 3.44. These stress results suggested that the beam was almost in pure bending.

For general nonlinear analysis including both geometric nonlinearity and elastic-plastic material properties, we considered a more practical problem involving collapse of a single stiffened panel. The 4-noded quad shell element mesh of this panel is shown in Figure 3.45. This panel, which contained geometric imperfections, was loaded in the axial direction. To ensure that all the nodes on the loaded end have identical axial displacements, multi-point constraints were specified in the X-direction. The master and slave nodes were also shown in Figure 3.45. However, when the Prefix.SMD file was generated by the direct VAST-DSA link, the multi-point constraints were applied to all six degrees of freedom between the master and slave nodes instead of only in the X-direction as in the original SMD file. This extension of the MPCs resulted in additional constraints on the deformation pattern at the loaded end, resulting in an increased stiffness in the pre-buckling range, as shown in Figure 3.46. The overall buckling of the stiffened panel was not dominated by this local deformation.

The deformed configurations, stress contours and plastic strain contours at various stages of the nonlinear deformation process obtained using the original Prefix.SMD file were presented in Figures 3.47-3.50. These results indicated that the local deformation of the stiffened panel at the loaded end was quite significant in the post-buckling regime. Tables for element nodal forces, nodal stresses and nodal plastic strains could also be displayed for user-defined element at user-selected solution steps, as shown in Figure 3.51.

The same nonlinear problem was also solved by the 3-noded triangular plate element using a mesh shown in Figure 3.52. The load-end shortening curves predicted by using this mesh were very close to those presented in Figure 3.46. The stress contour plot for a solution step shortly after the peak load is given in Figure 3.53, which is in good agreement with the middle plot shown in Figure 4.49.

For nonlinear analysis involving large strain deformation of hyper-elastic material, we considered indentation of a rubber block by a rigid surface. This test problem was solved using the large strain version of the 20-noded solid element, IEC52. The original and deformed meshes are presented in Figure 3.54. The rubber block was confined in a box and the displacements in the direction normal to the plane of paper were constrained to achieve a plane-strain condition. So only the upper surface of the rubber block was free to deform. In the deformed shape shown in Figure 3.54, the indentation of the rigid surface caused a 50% reduction on the original height of the rubber block and caused extremely large strains inside the body. The stress contour at the final load level is presented in Figure 3.55.

3.1.3 Dynamic Analysis

For dynamic analysis, we first considered linear elastic time history analysis through direct time integration using a simply supported plate as a test example. This plate was initially at rest and subjected to a sudden application of uniform pressure at time zero. The 4-noded quad shell elements were employed. The predicted deformed configuration at a typical time step and the time history of center deflection are given in Figures 3.56 and 3.57, respectively. The stress contours at different time steps are presented in Figures 3.58 and 3.59. Although the magnitudes of stresses were different in these figures, the patterns of the contours were very similar. This was expected because this was a linear problem. Similar to the static analyses considered before, tables for element nodal stresses and nodal forces could also be generated as shown in Figures 3.60 and 3.61.

For nonlinear dynamic analysis, we considered the geometrically nonlinear version of the same test case, where the predicted deformed shape and time history of center deflection are presented in Figures 3.62 and 3.63. Although the general deformation pattern for the nonlinear case was similar to its linear counterpart, the peak displacements predicted in the nonlinear solution were significantly smaller than those in the linear solution and the period in the nonlinear solution was also much shorter than that in the linear solution. This was due to the stress-hardening effect introduced by geometric nonlinearity. The stress contours at typical time steps were displayed in Figures 3.64 and 3.65. The patterns of stress contours are now varying because of nonlinearity in the problem. Tables for nodal stresses and nodal forces could also be generated as shown in Figures 3.66 and 3.67.

3.1.4 Eigenvalue Analysis

To verify the capability of the direct VAST-DSA link for formulating input data files for natural frequency analysis and post-processing modal stresses, we again employed the model of simply supported plate. The first five modes of free vibration and the corresponding modal stresses were computed using the direct VAST-DSA link. The calculated mode shapes and contours of modal stresses for the first two modes are predicted in Figures 3.68-3.71. The stress plots were verified by transferring stress data from the V53 file to the DSA database and also by comparing with the Prefix.LPT file.

In the final example problem, the “dry” and “wet” modes of a cylinder in water were calculated using the direct VAST-DSA link. This test example was utilized to verify the pre-processing feature for the fluid added mass information provided in the Prefix.AMD file. The AMD file generated during a VAST execution was compared with the existing AMD file and they were found to be identical. The “dry” and “wet” modes computed using the direct VAST-DSA link were also identical with those obtained using the original set of ASCII input data file. The original finite element mesh including the fluid elements are shown in Figure 3.72 and a typical “wet” mode is given in Figure 3.73. Stress contour plot for the tenth mode is given in Figure 3.74.

3.2 Benchmark of Numerical Performance

The implementation and verification of the direct linkage between the DSA database and VAST program for pre- and post-processing were in the preceding sections in this report. In the present section, we present a benchmark study to investigate the efficiency of the direct DSA-VAST link for post-processing stresses and nodal forces for large finite element models with large numbers of load cases.

The example problem utilized in this benchmark study was the global finite element model of a CPF with a total of 207 load cases, including rigid body motions, unit panel pressures and still-water bending moment. The specifications of this finite element model and the load cases are presented in Figures 3.19 and 3.20.

For this test example, significant differences were noticed between the elapsed times taken by the STRESS module when different options were used in the VAST executions for treating the stress results. A comparison of the run times is presented in Table 3.1. As indicated by these results, if the internal force vectors were not computed and stored, the population of the VES file in the DSA database is almost as efficient as the generation of the V53 file. On the other hand, the elapsed times

required to create the VES and V53 files are also close if there is only one load case. However, when the internal force vectors and nodal stresses were both selected for a model with a large number of load cases, the population of the VES, VNS and VEF files in DSA database took a much longer time than required to generate the V53 file. The sizes of the DSA database files are considerably smaller than the V53 file, as indicated in Table 3.2, as the DSA files contained less information. For example, the VES file only contained stresses at the element center.

Although the benchmark results suggested that the direct DSA-VAST linkage reduced the speed of VAST execution for large finite element models with large numbers of load cases, the advantage for having such a direct linkage is still obvious. It integrates the step for generating input data files and populating the stress files in the DSA database into VAST execution, which is normally performed in a batch mode, so the need for the interactive “stress transfer” in the GUI program is eliminated. Actually, the benchmark data in Table 3.1 suggested that the stress transfer capability in the Trident GUI is even less efficient than that in the VAST solver.

Table 3.1: Comparison of Execution Times for the CPF Model

Program Module Name	Files Generated	Data Included	Elapsed Time (s)	
			1 load case	207 load cases
VAST (Stress)	V53	Element stresses	55.390	795.474
VAST (Stress)	DSA	Element stresses	55.015	872.222
VAST (Stress)	V53	Element stresses, Nodal forces	57.734	855.628
VAST (Stress)	DSA	Element stresses, Nodal forces, Nodal stresses	58.765	4862.235
Trident FEA	V53→DSA	Element stresses, Nodal forces, Nodal stresses	6.000	6120.000

Table 3.2: Summary of Sizes of Stress Files Generated by VAST and DSA Database for the CPF Model

File Extension	Descriptions	Size (KB)	
		1 load case	207 load cases
V53	VAST result file for stresses without nodal forces	16,722	3,825,339
V53	VAST result file for stresses with nodal forces	19,351	3,281,139
VES	DSA Database file for element stresses	2,686	279,241
VNS	DSA Database file for nodal stresses	7,653	795,835
VEF	DSA Database file for element nodal forces	10,472	1,089,037

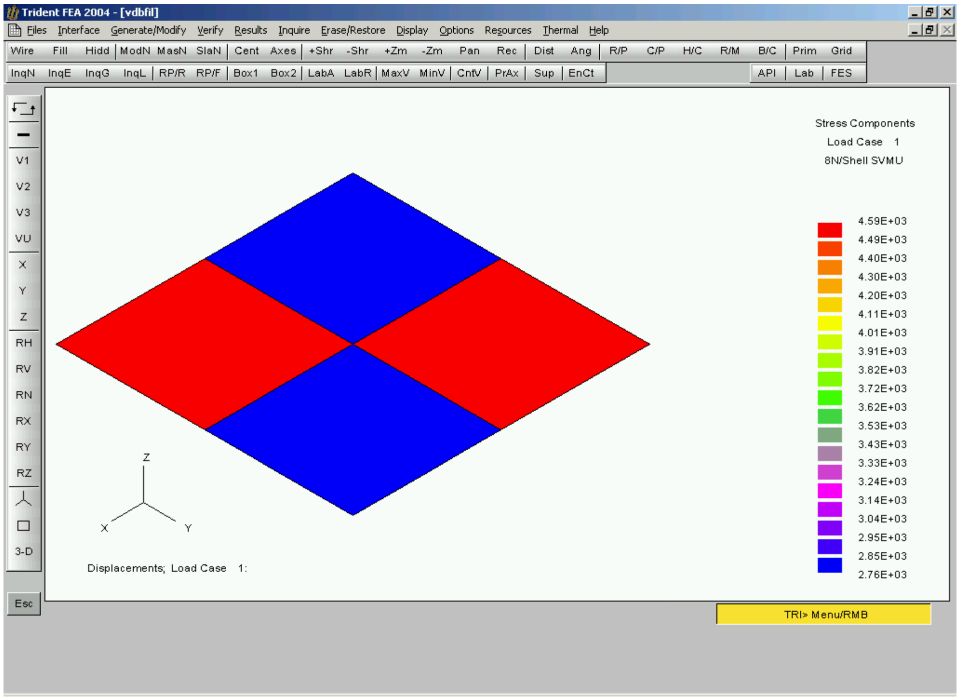


Figure 3.1: Element Stress Plot for Linear Static Analysis of a Simply Supported Plate Modeled by 8-Noded Shell Elements.

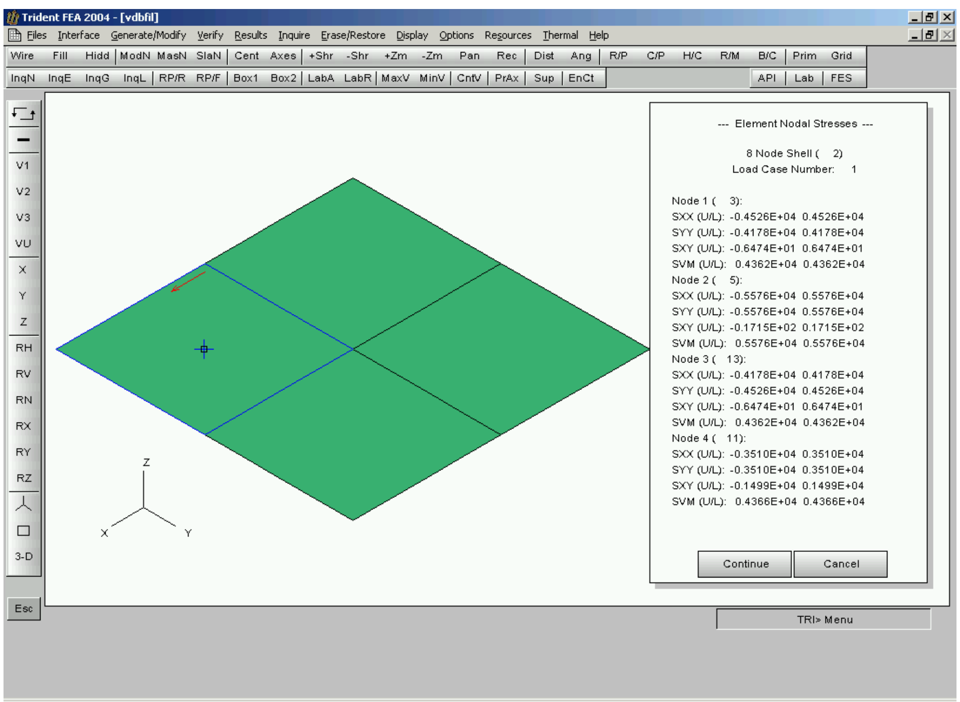


Figure 3.2: Nodal Stresses in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 8-Noded Shell Elements.

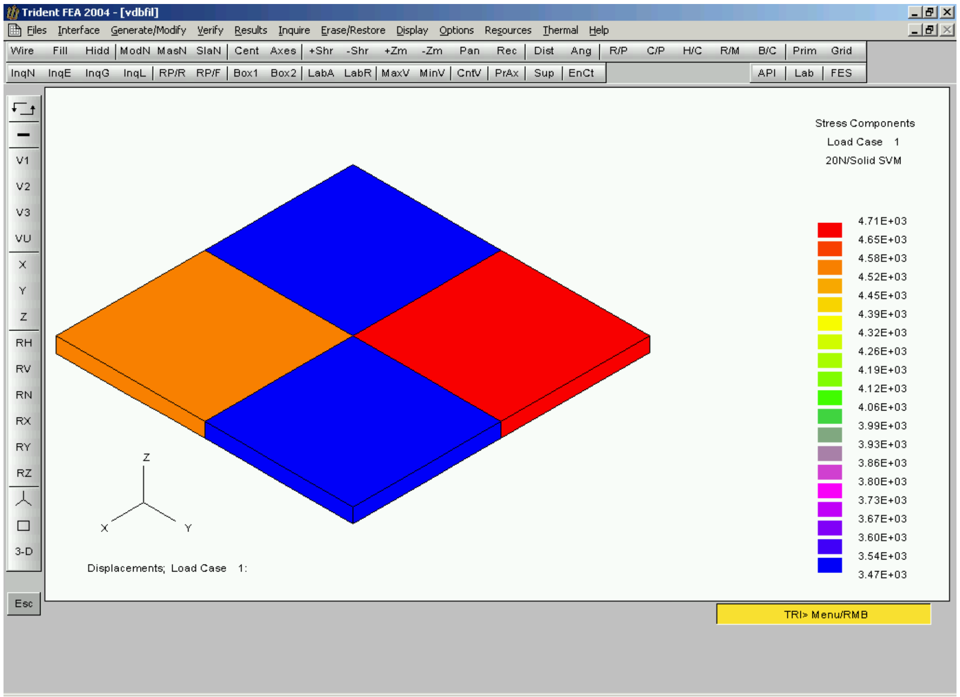


Figure 3.3: Element Stress Plot for Linear Static Analysis of a Simply Supported Plate Modeled by 20-Noded Solid Elements.

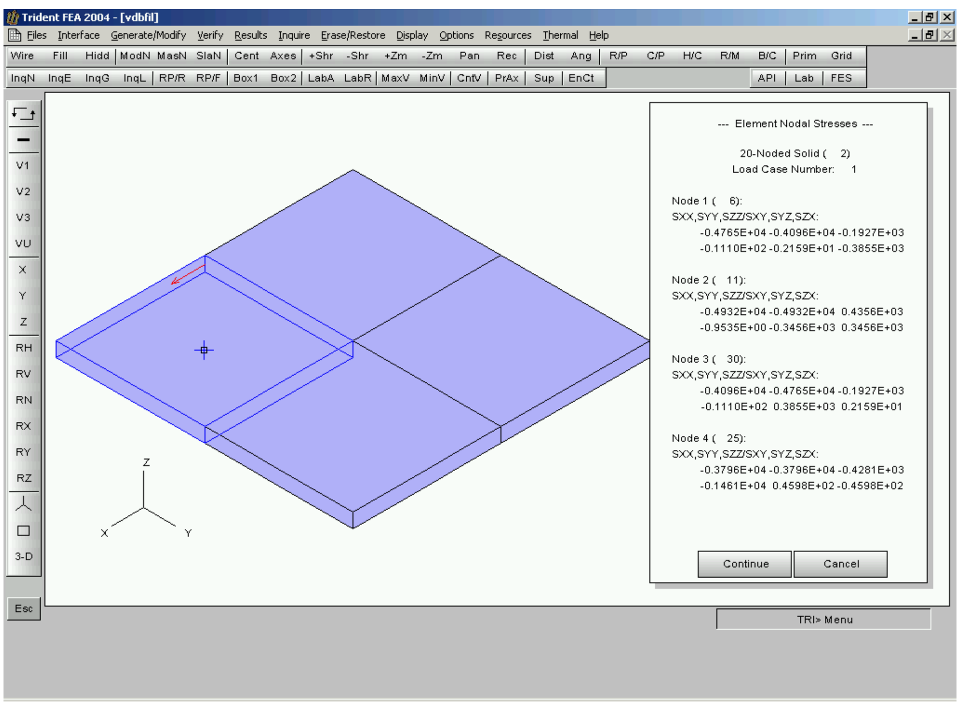


Figure 3.4: Nodal Stresses in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 20-Noded Solid Elements.

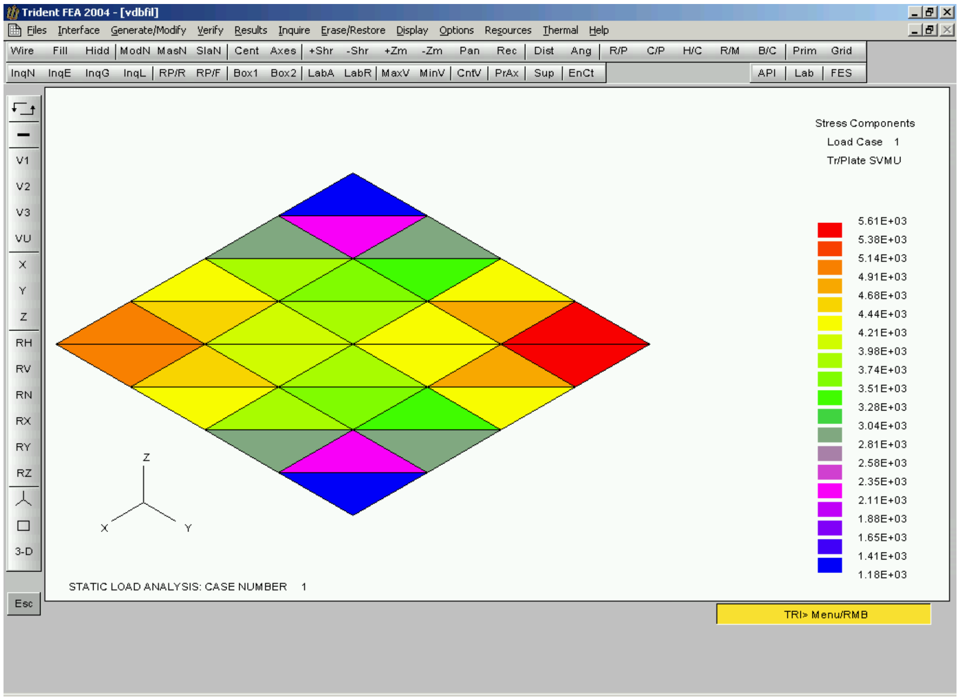


Figure 3.5: Element Stress Plot for Linear Static Analysis of a Simply Supported Plate Modeled by 3-Noded Plate Elements.

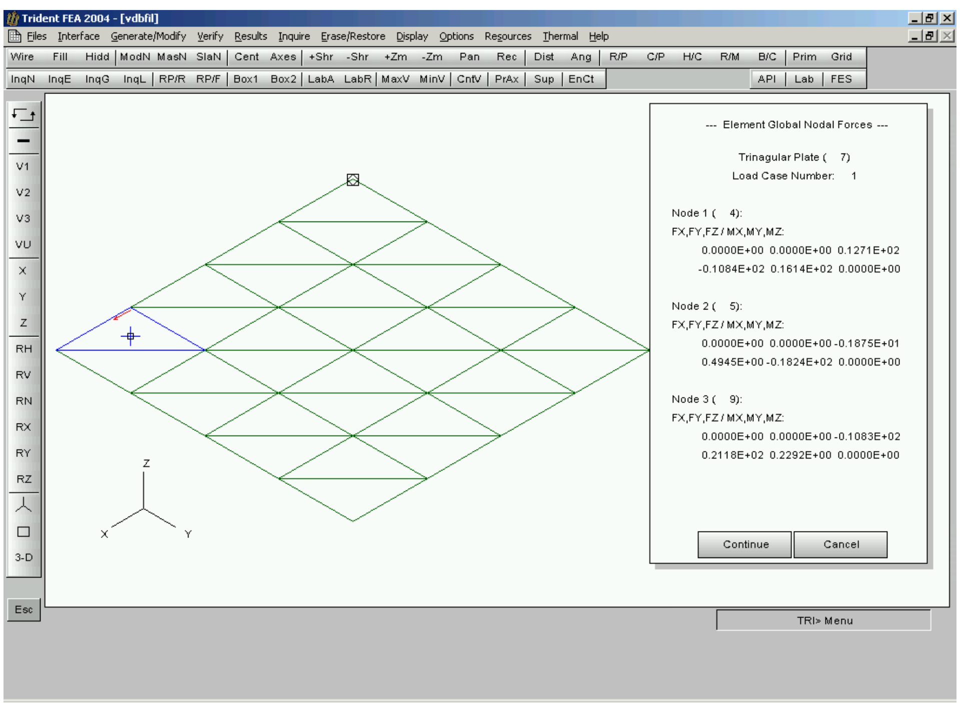


Figure 3.6: Nodal Forces in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 3-Noded Plate Elements.

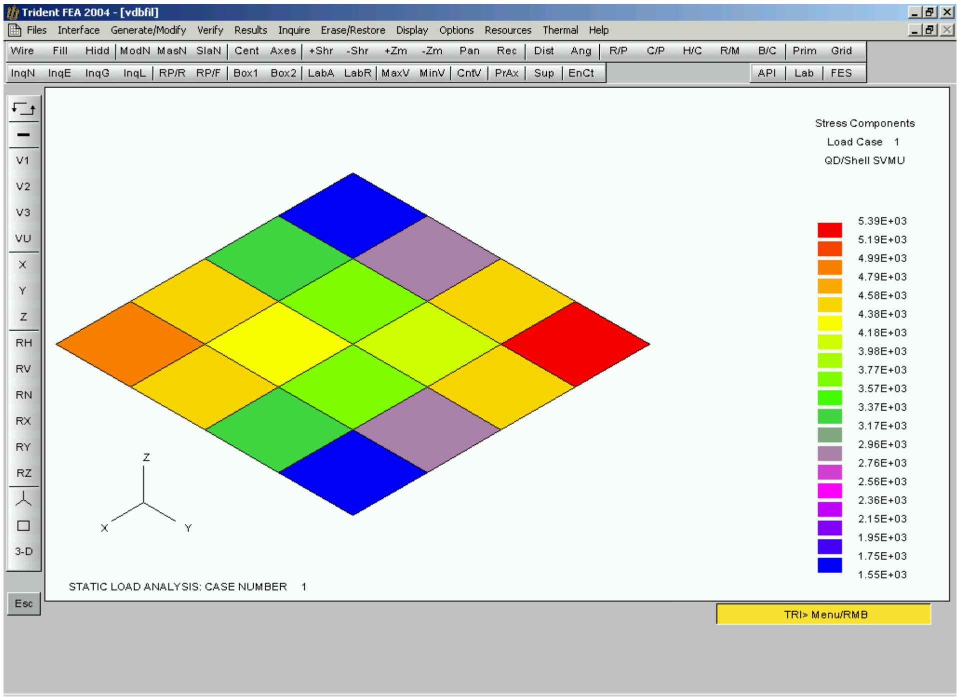


Figure 3.7: Element Stress Plot for Linear Static Analysis of a Simply Supported Plate Modeled by 4-Noded Shell Elements.

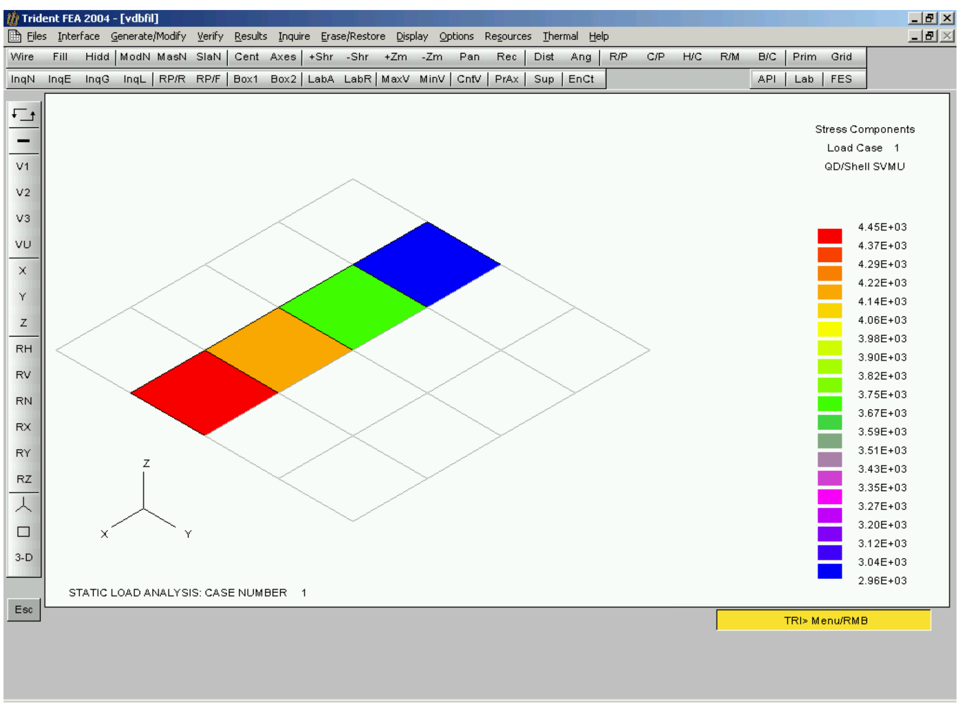


Figure 3.8: Element Stress Plot for the Simply Supported Plate Modeled by 4-Noded Shell Elements When Stresses Were Only Computed in Selected Elements.

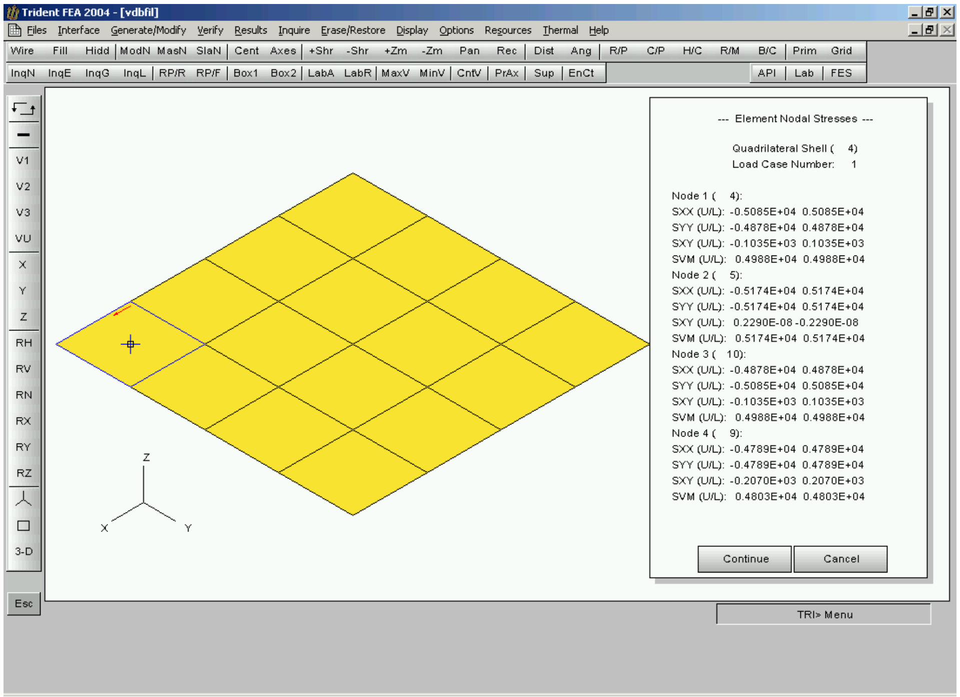


Figure 3.9: Nodal Stresses in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 4-Noded Shell Elements.

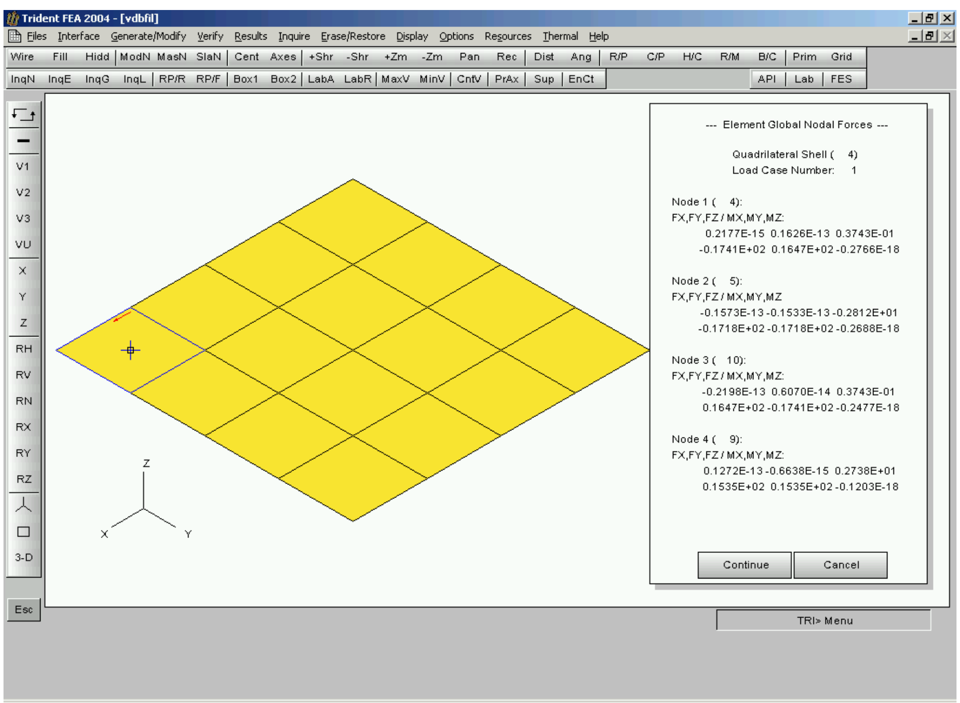


Figure 3.10: Nodal Forces in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 4-Noded Shell Elements.

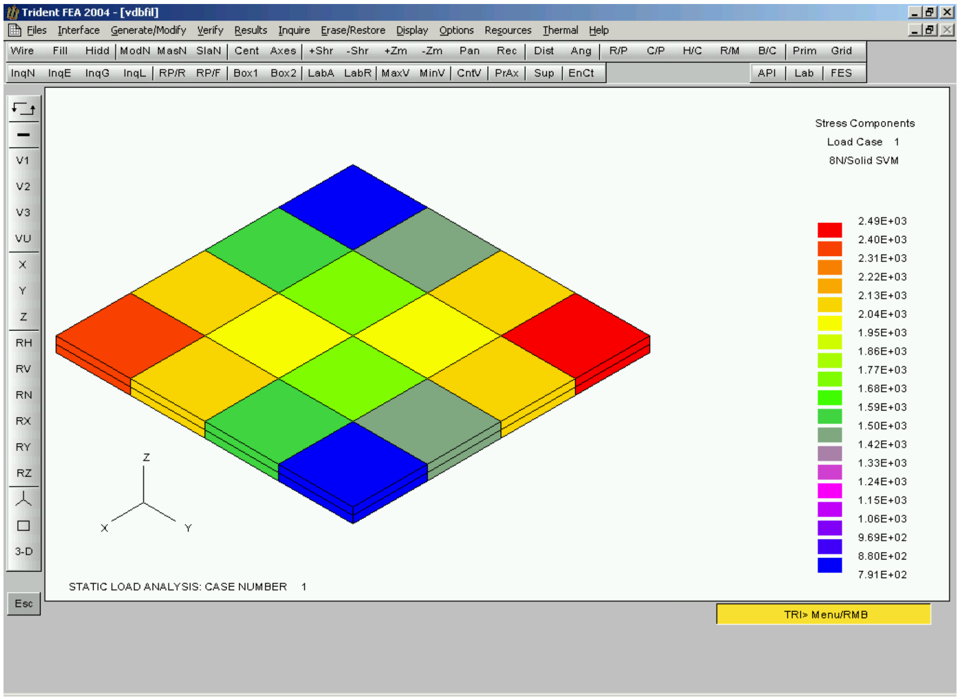


Figure 3.11: Element Stress Plot for Linear Static Analysis of a Simply Supported Plate Modeled by 8-Noded Solid Elements.

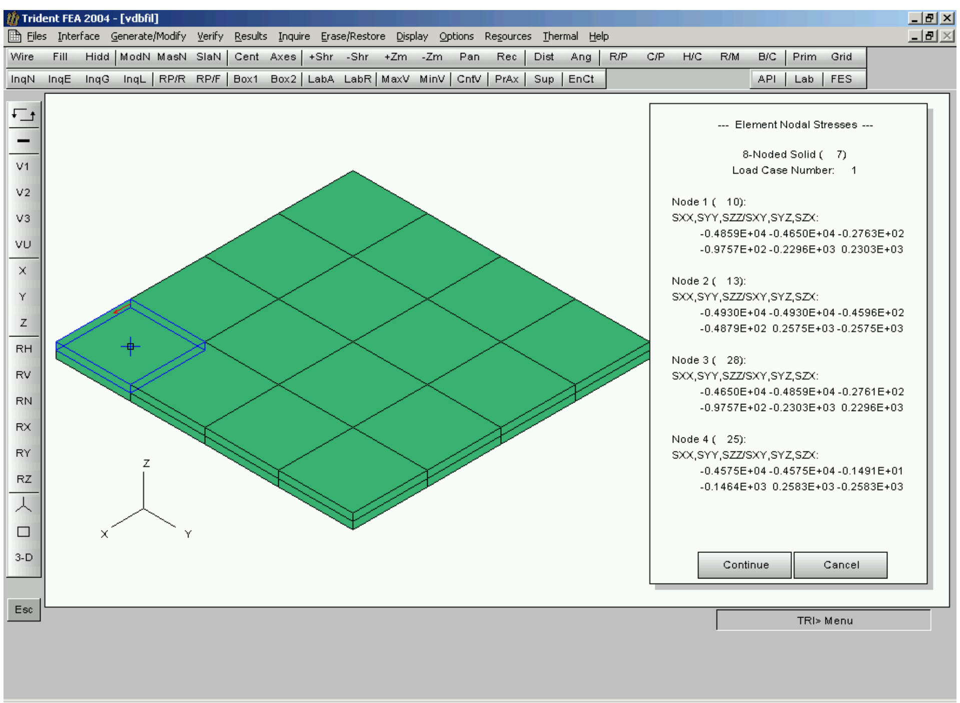


Figure 3.12: Nodal Stresses in Selected Element for Linear Static Analysis of a Simply Supported Plate Modeled by 8-Noded Solid Elements.

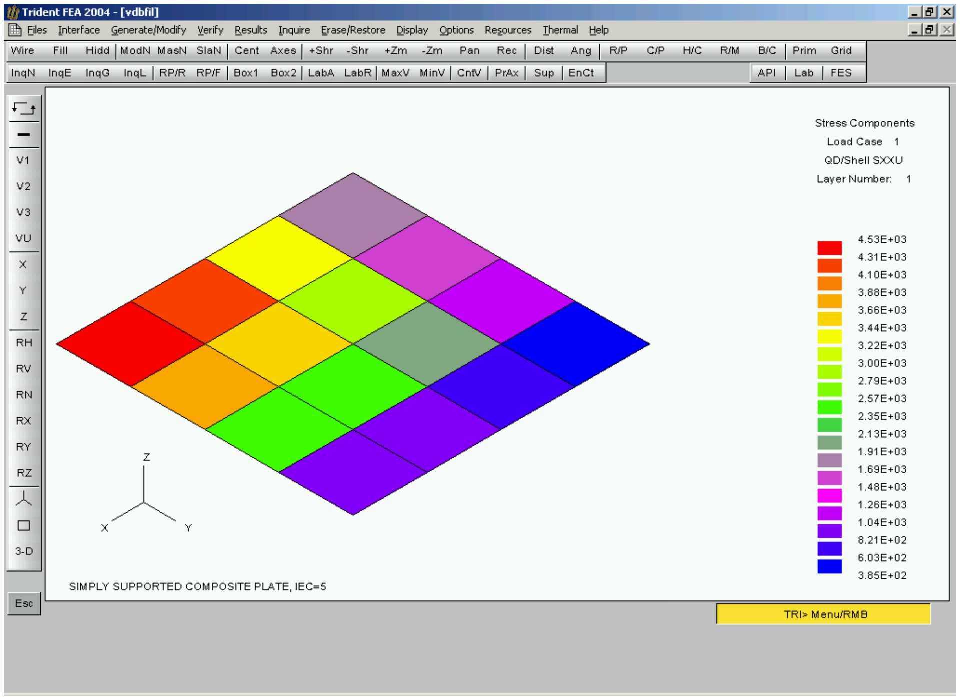


Figure 3.13: Element Stress Plot on the Upper Surface of the First Layer for Linear Static Analysis of a Simply Supported Composite Plate.

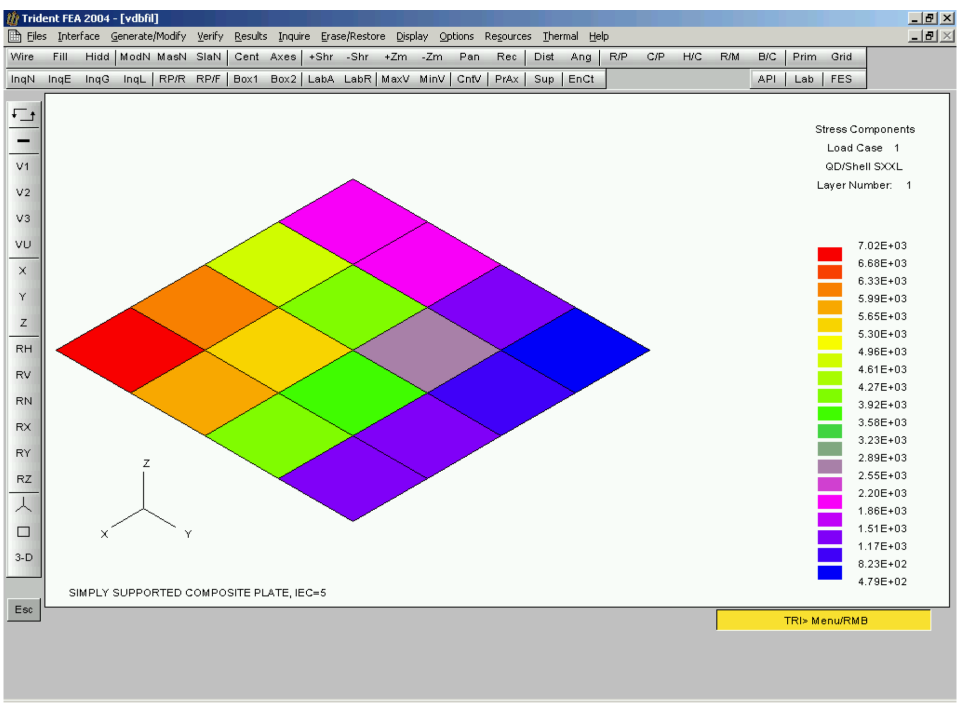


Figure 3.14: Element Stress Plot on the Lower Surface of the First Layer for Linear Static Analysis of a Simply Supported Composite Plate.

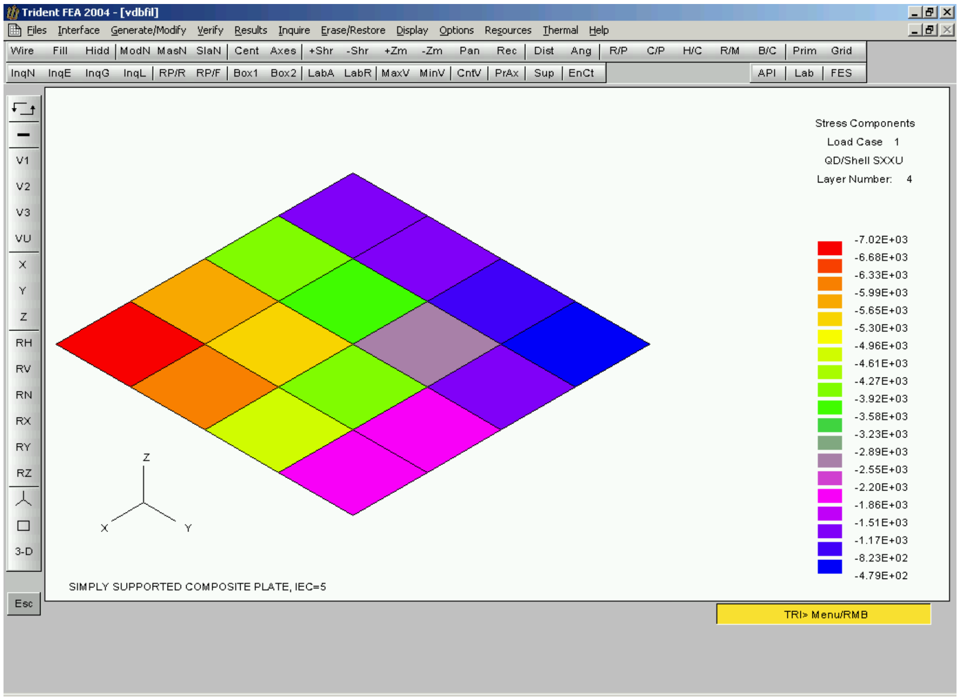


Figure 3.15: Element Stress Plot on the Upper Surface of the Fourth Layer for Linear Static Analysis of a Simply Supported Composite Plate.

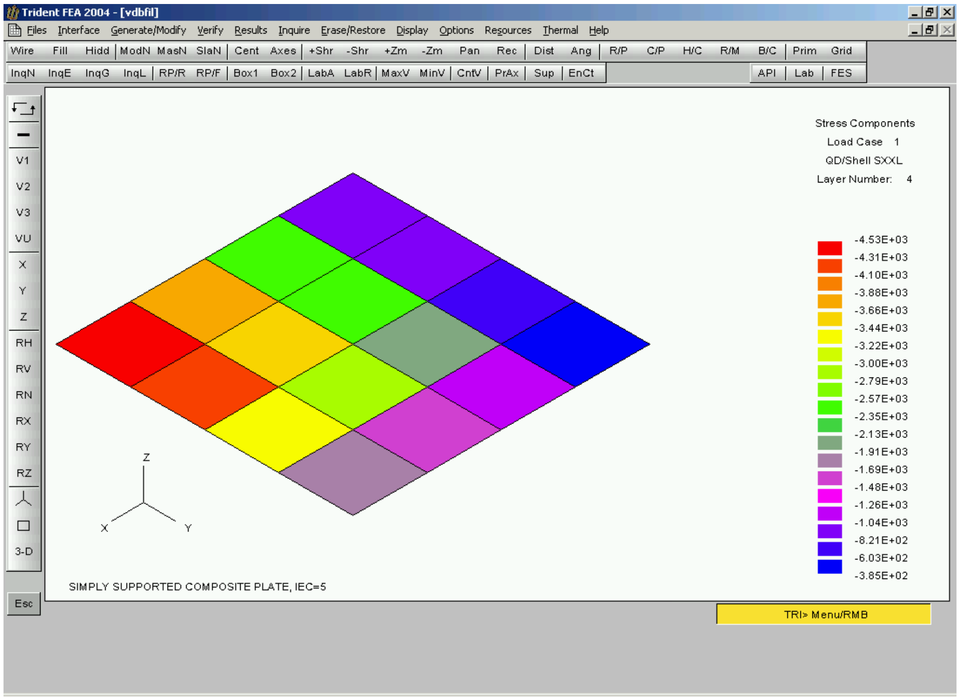


Figure 3.16: Element Stress Plot on the Lower Surface of the Fourth Layer for Linear Static Analysis of a Simply Supported Composite Plate.

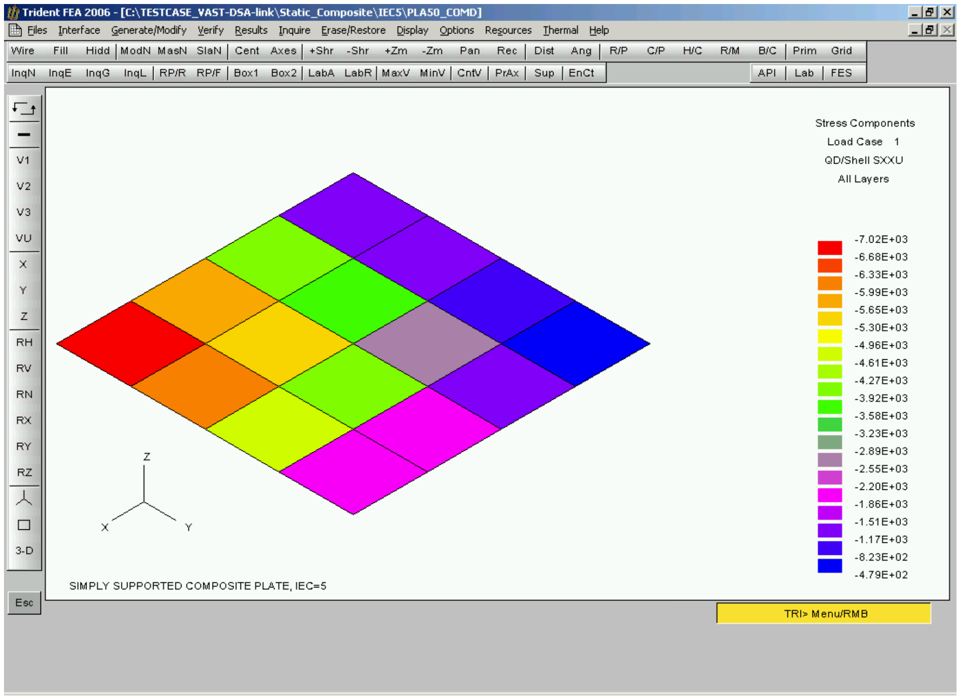


Figure 3.17: Element Stress Plot on the Upper Surface of the Whole Plate for Linear Static Analysis of a Simply Supported Composite Plate.

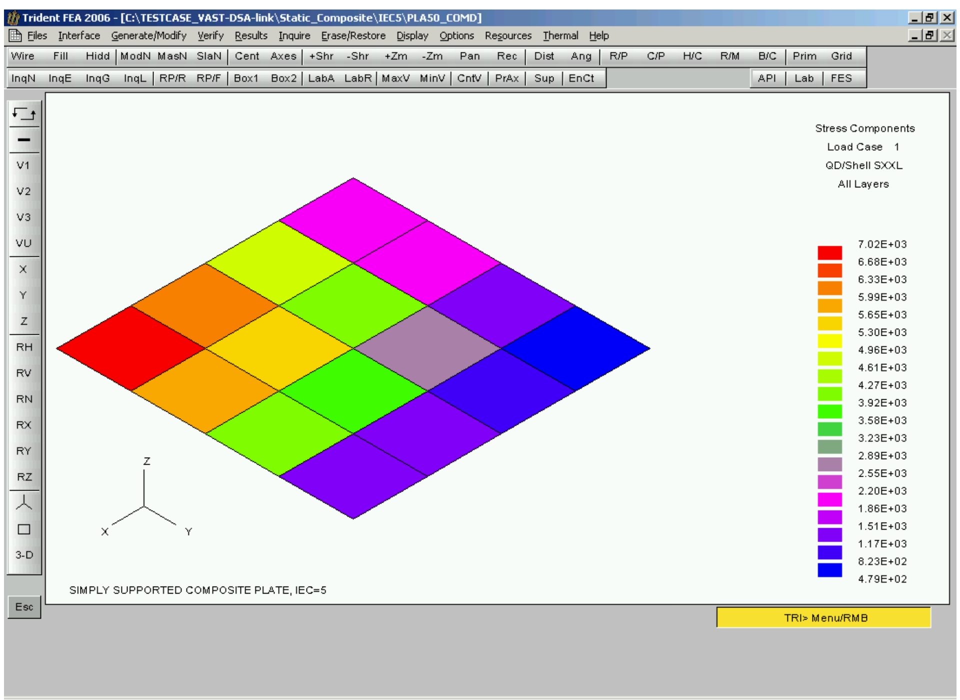


Figure 3.18: Element Stress Plot on the Lower Surface of the Whole Plate for Linear Static Analysis of a Simply Supported Composite Plate.

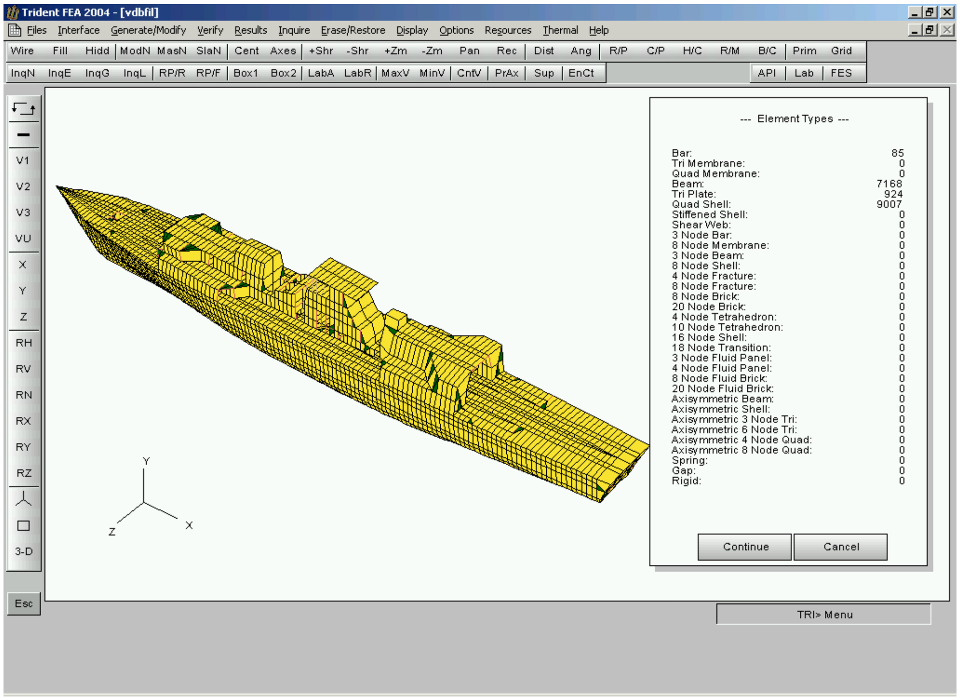


Figure 3.19: Summary of Finite Element Mesh for Linear Static Analysis of a Global CPF Model Involving 207 Load Cases.

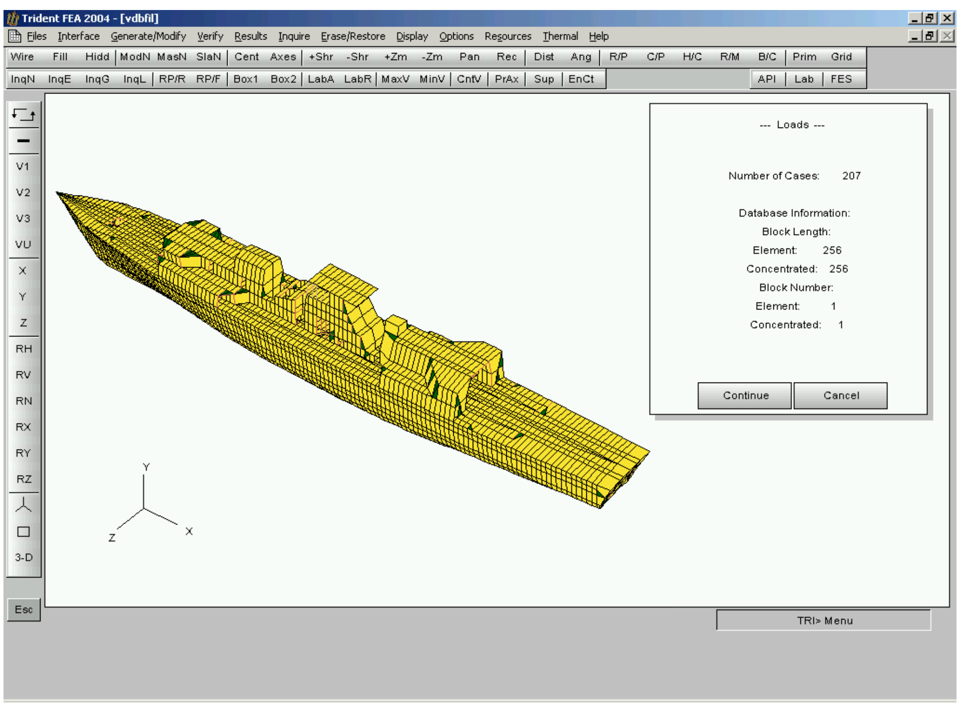


Figure 3.20: Summary of Load Cases for Linear Static Analysis of a Global CPF Model.

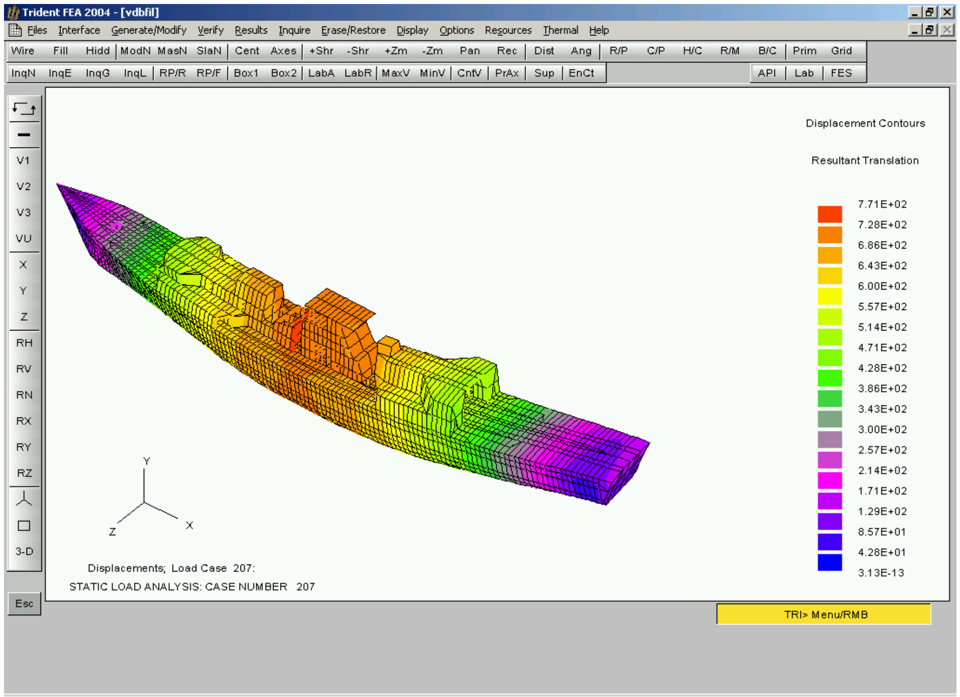


Figure 3.21: Deformed Shape and Displacement Contour of the Global CPF Model for the Final (207th) Load Case.

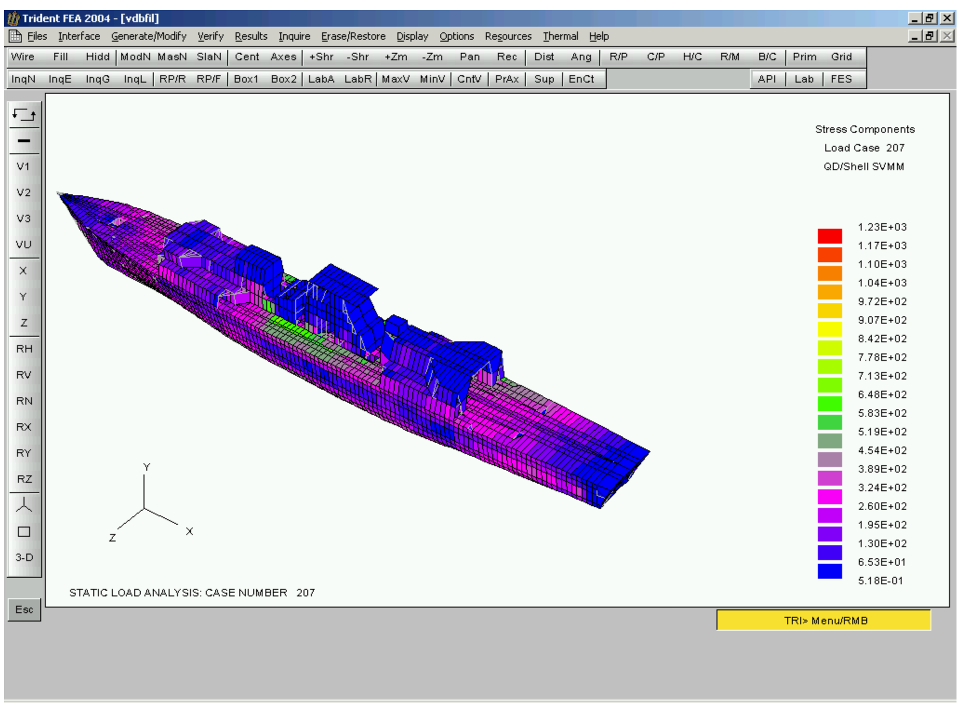


Figure 3.22: Stress Contour of the Global CPF Model for the Final (207th) Load Case.

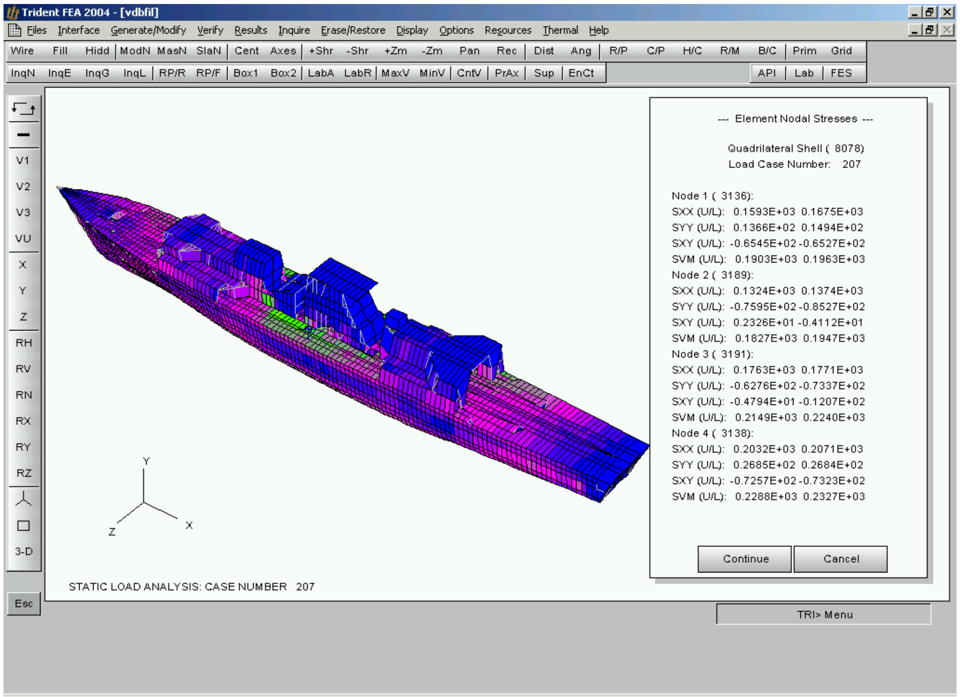


Figure 3.23: Nodal Stresses in a User-Selected Element on the Main Deck in the Global CPF Model for the Final (207th) Load Case.

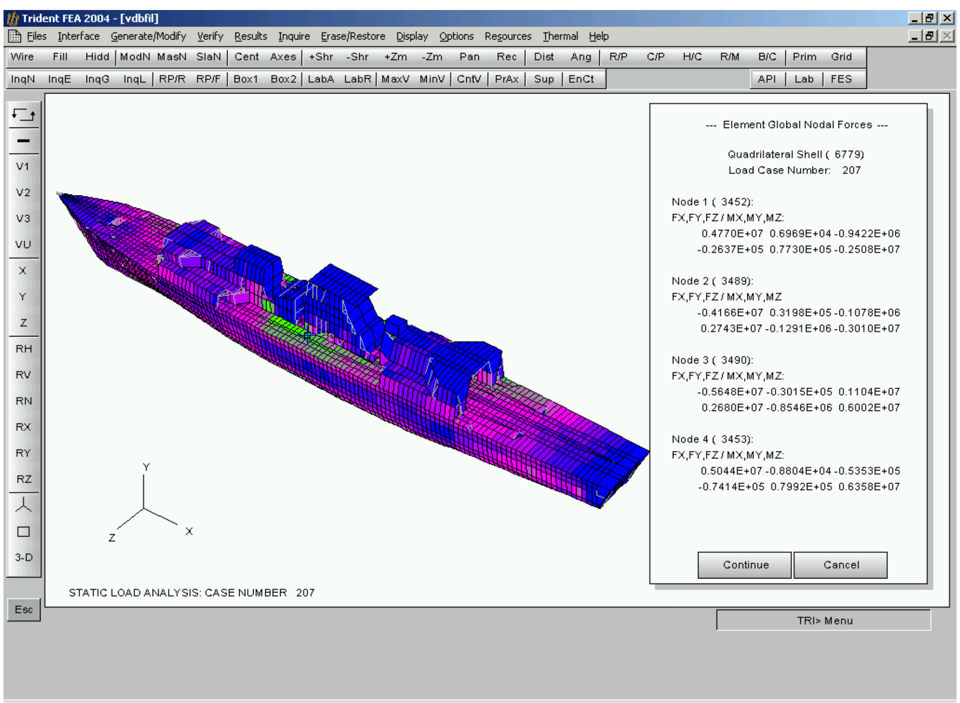


Figure 3.24: Nodal Forces in a User-Selected Element on the Main Deck in the Global CPF Model for the Final (207th) Load Case.

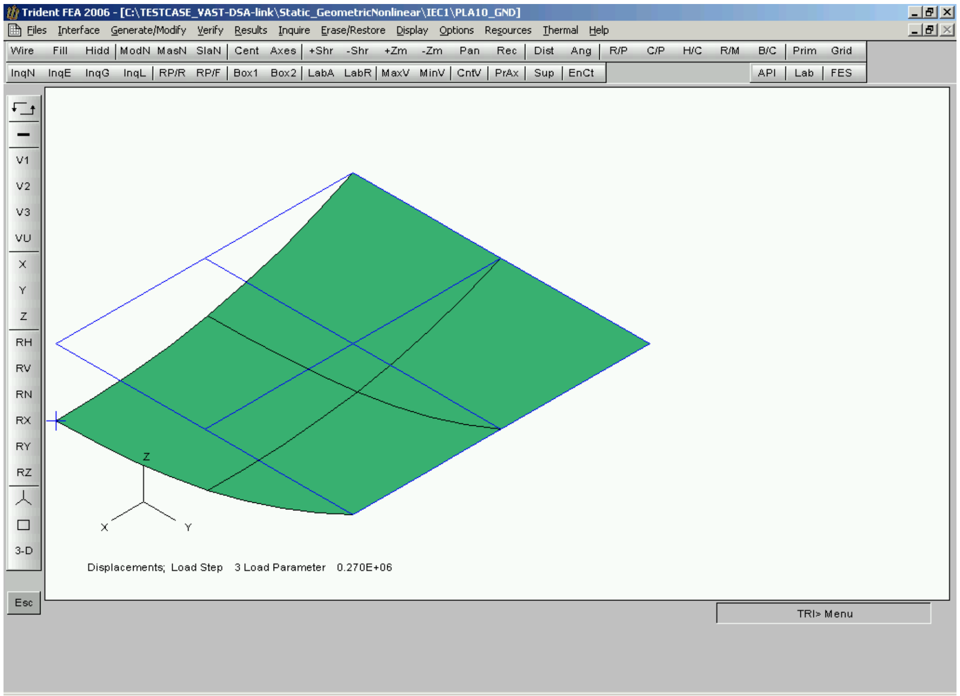


Figure 3.25: Geometrically Nonlinear Deformed Shape of a Simply Supported Plate Modelled by 8-Noded Shell Elements.

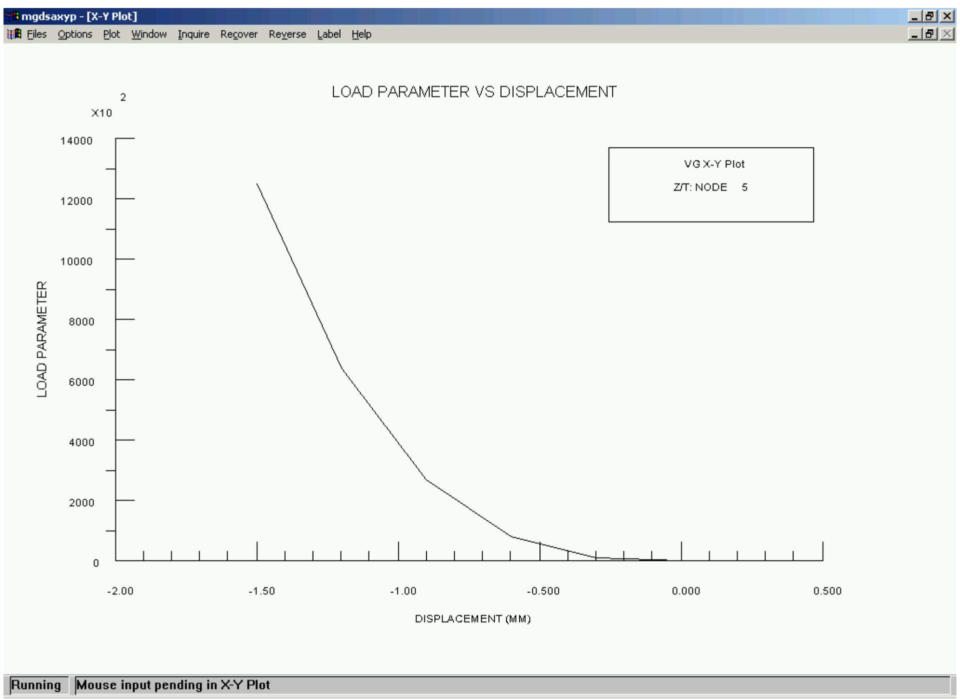


Figure 3.26: Load-Center Deflection Curve for Geometrically Nonlinear Analysis of a Simply Supported Plate Obtained Using 8-Noded Shell Elements.

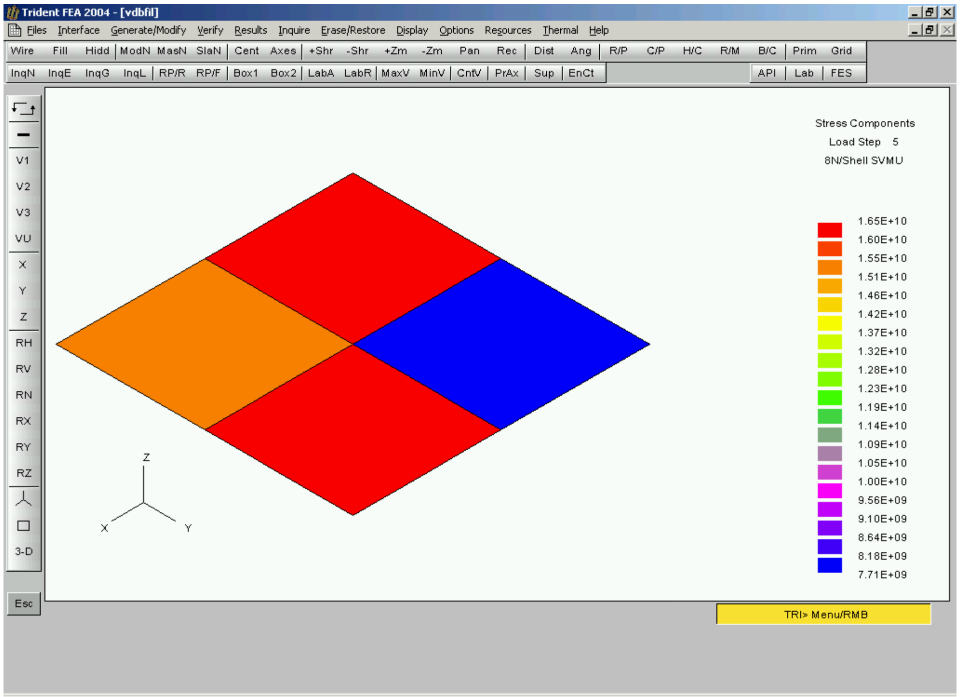


Figure 3.27: Element Stress Plot at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 8-Noded Shell Elements.

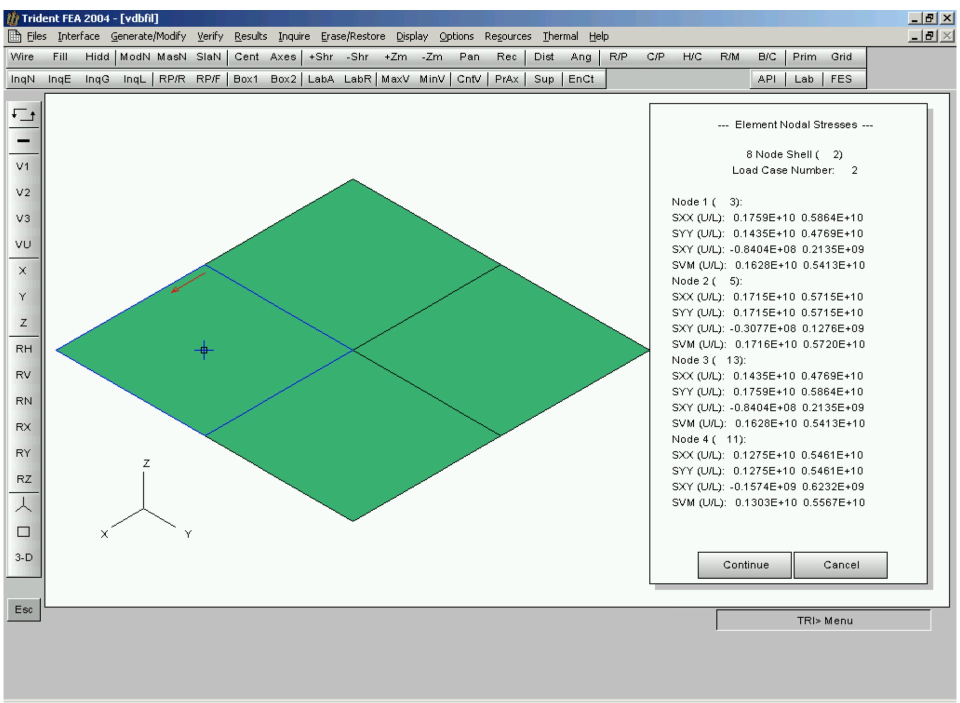


Figure 3.28: Nodal Stresses in Element #2 at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 8-Noded Shell Elements.

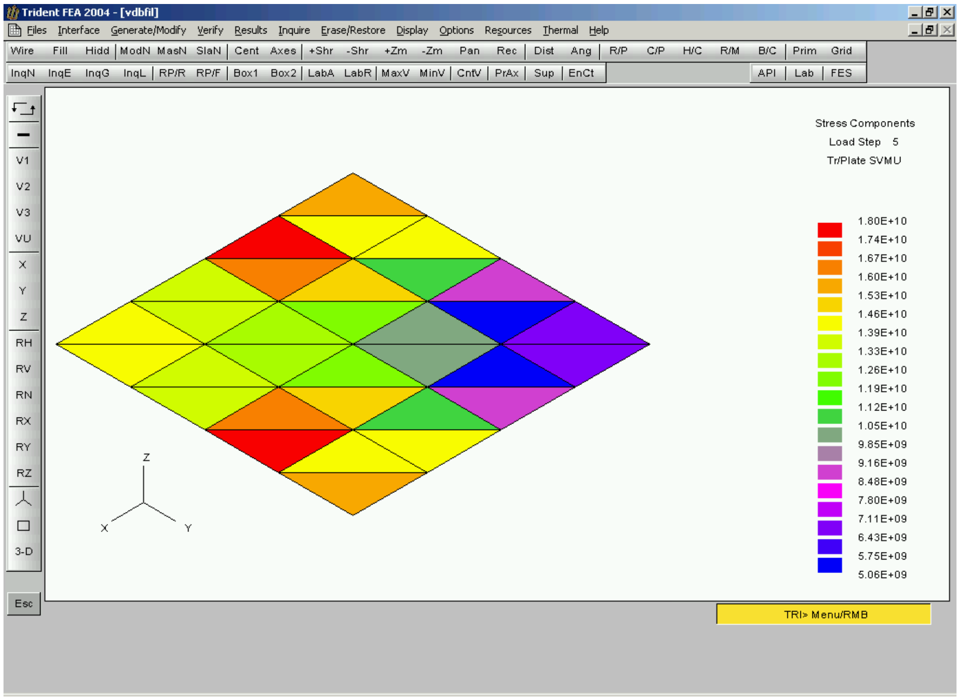


Figure 3.29: Element Stress Plot at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 3-Noded Plate Elements.

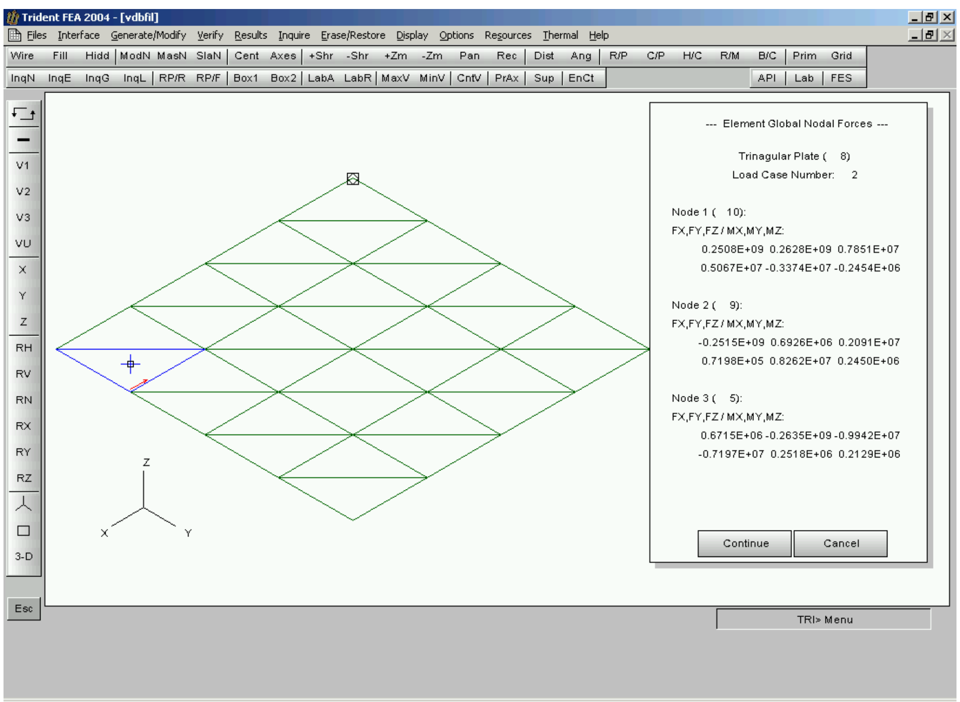


Figure 3.30: Nodal Forces in Element #8 at Solution Step 2 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 3-Noded Plate Elements.

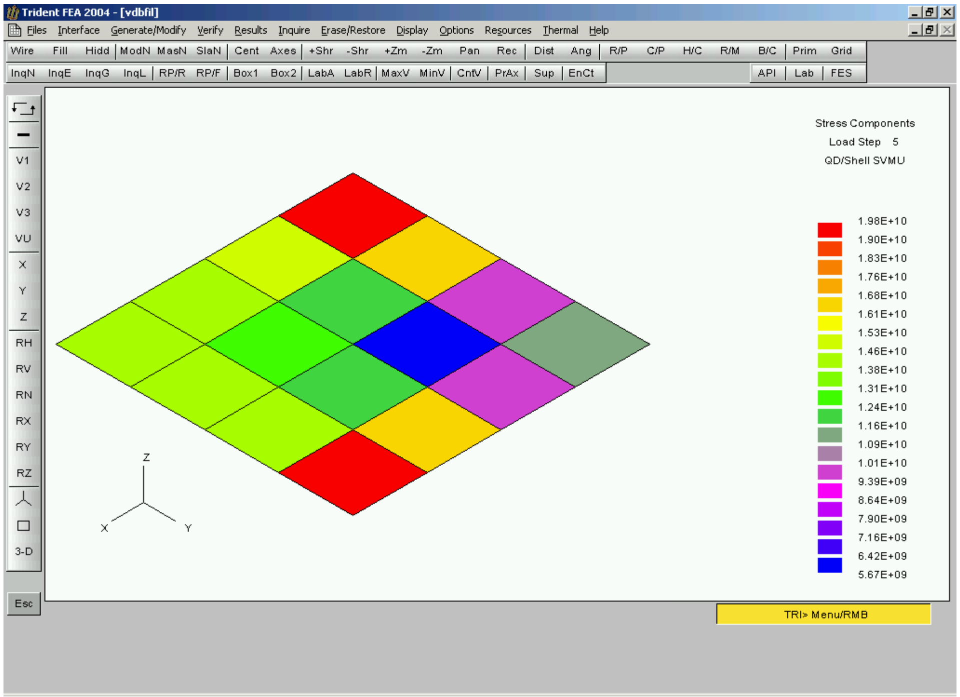


Figure 3.31: Element Stress Plot at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 4-Noded Quad Shell Elements.

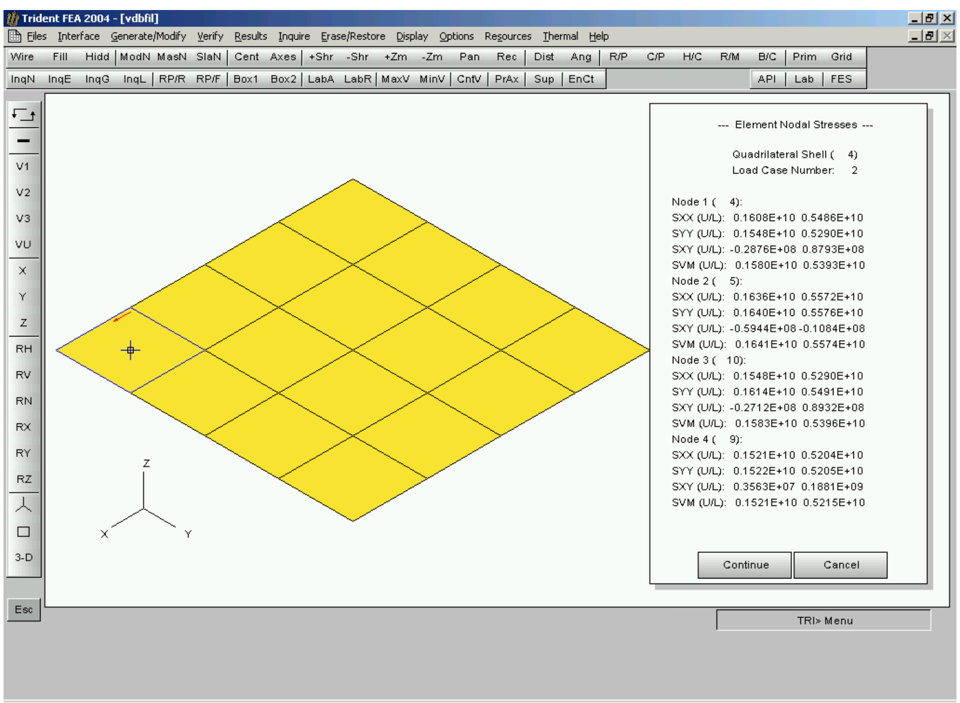


Figure 3.32: Nodal Stresses in Element #4 at Solution Step 2 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 4-Noded Shell Elements.

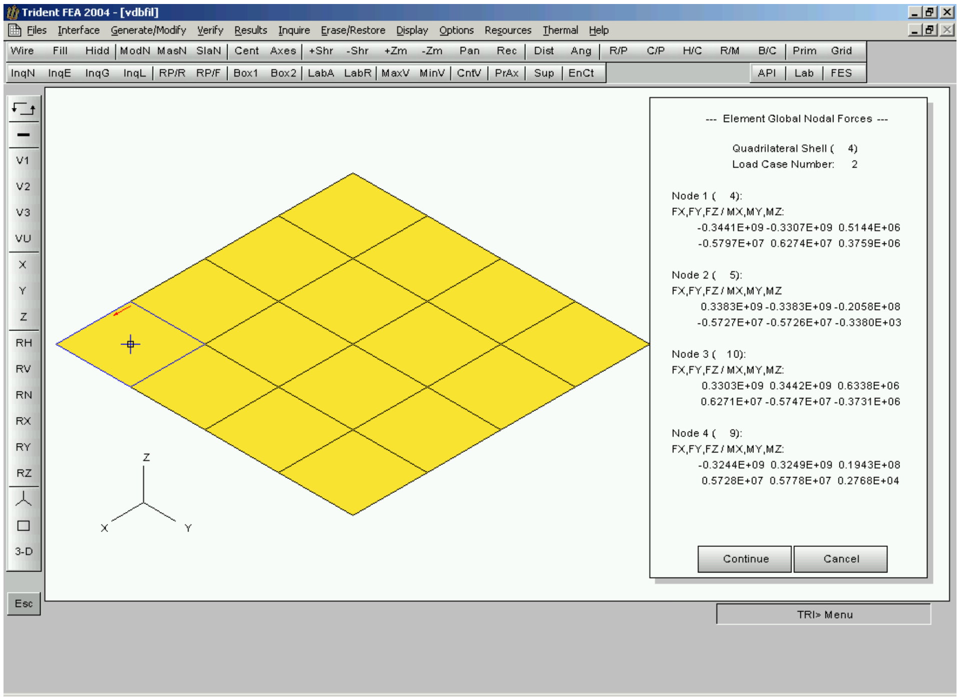


Figure 3.33: Nodal Forces in Element #4 at Solution Step 2 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 4-Noded Shell Elements.

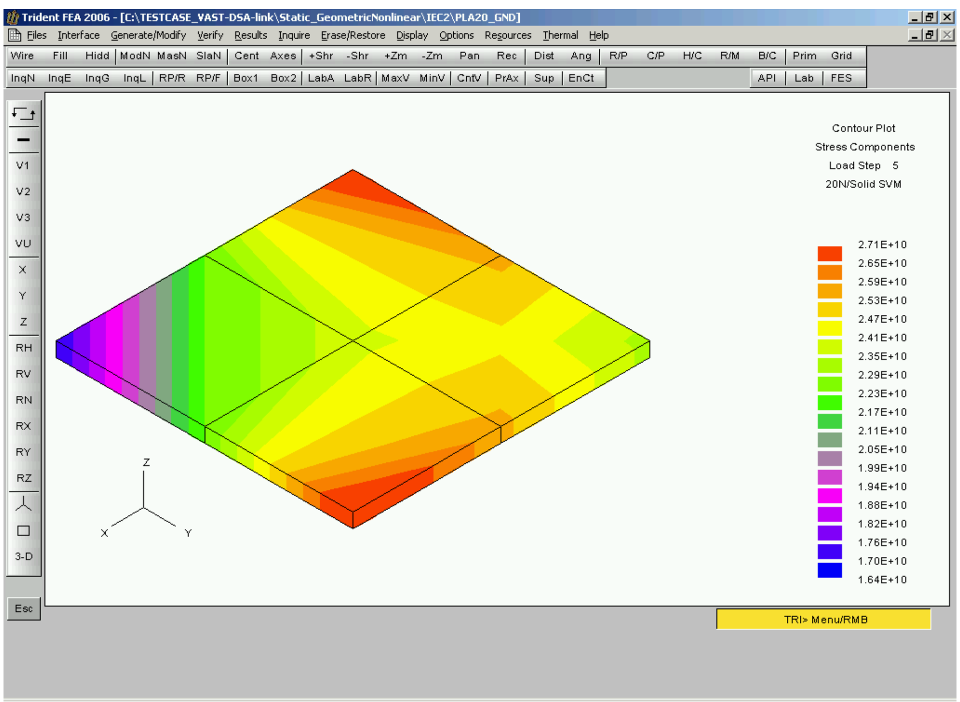


Figure 3.34: Stress Contour at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 20-Noded Solid Elements.

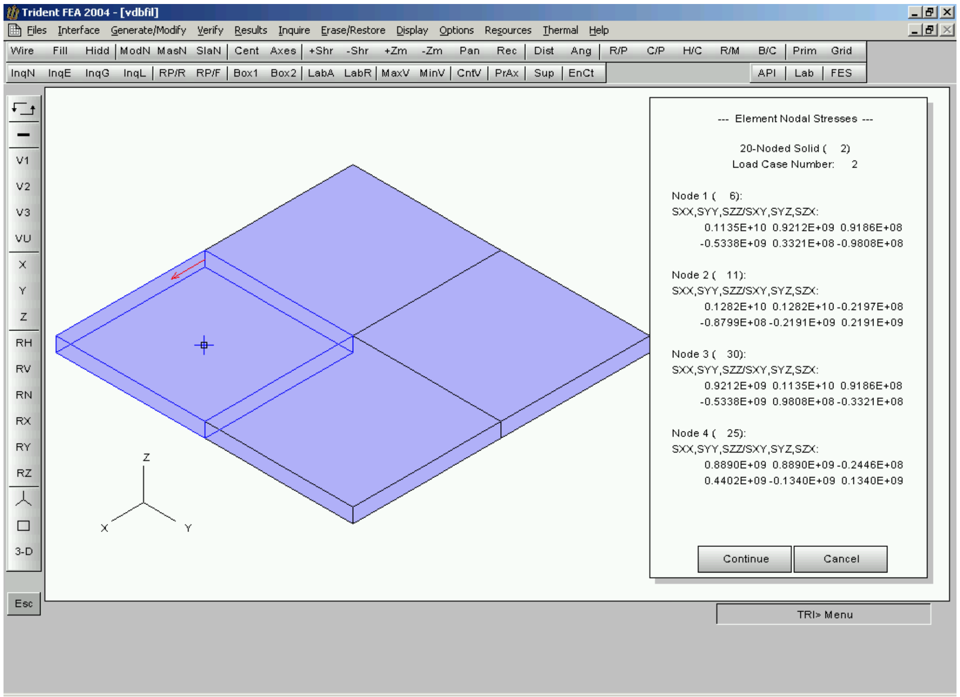


Figure 3.35: Nodal Stresses on Upper Surface of Element #2 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 20-Noded Solid Elements.

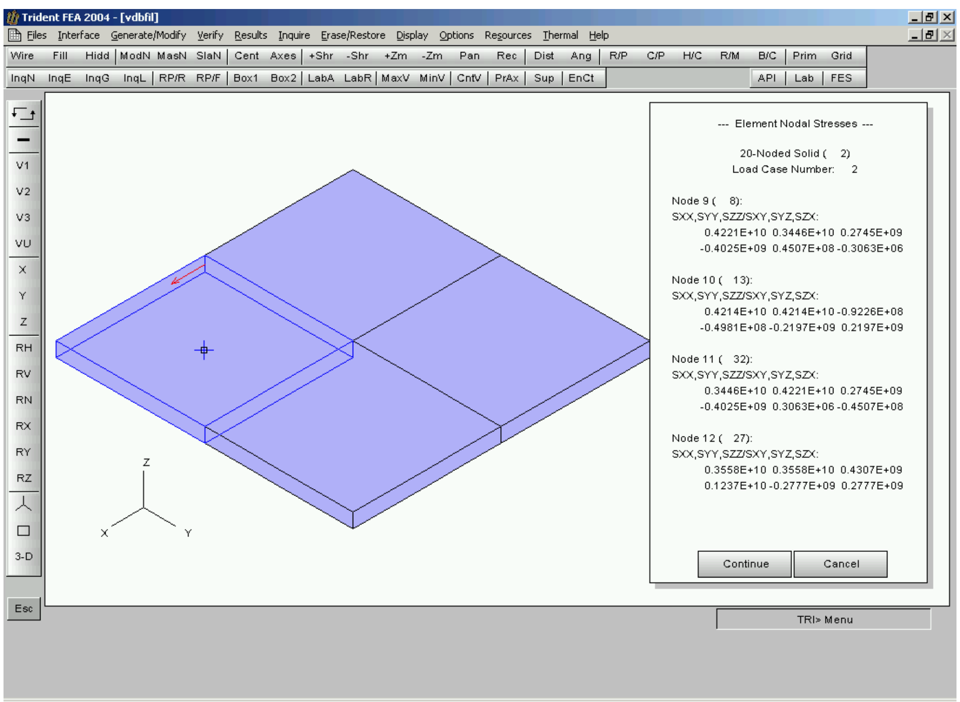


Figure 3.36: Nodal Stresses on Lower Surface of Element #2 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 20-Noded Solid Elements.

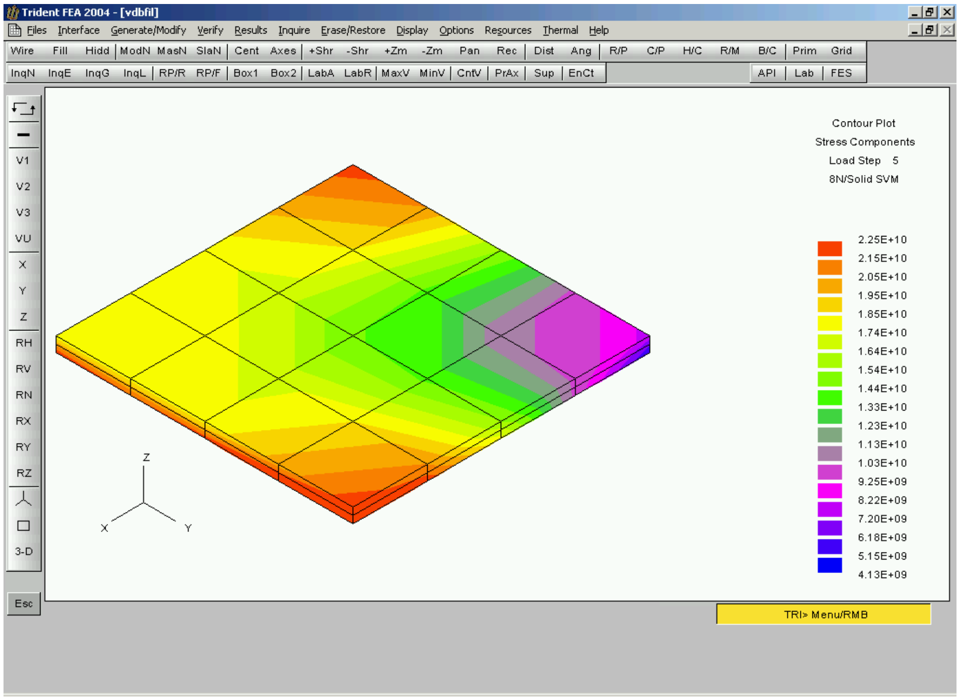


Figure 3.37: Stress Contour at Solution Step 5 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 8-Noded Solid Elements.

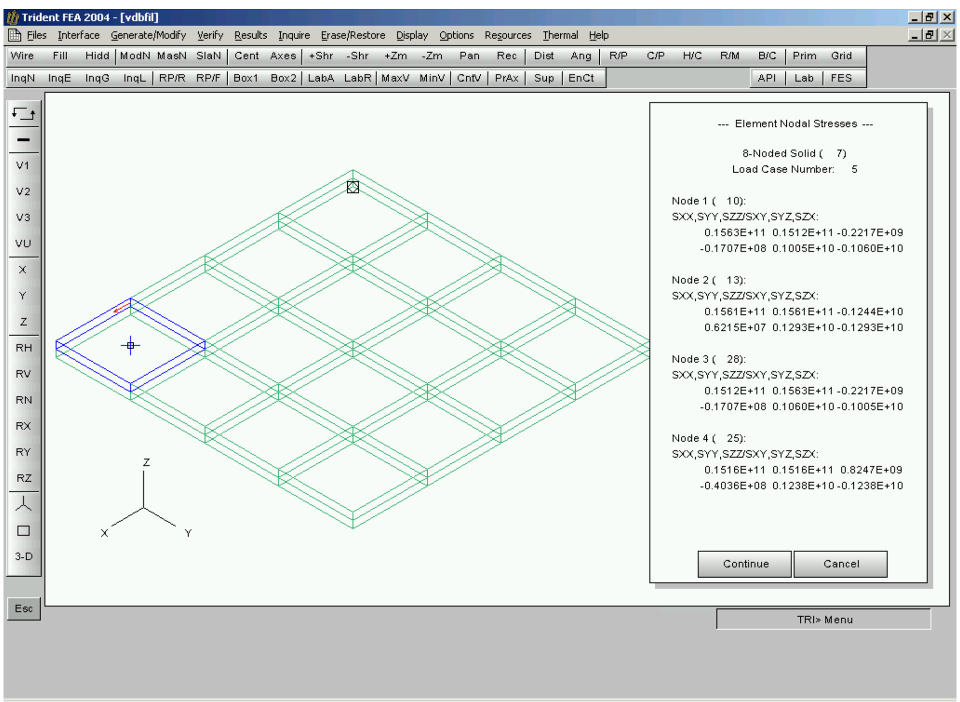


Figure 3.38: Nodal Stresses on Upper Surface of Element #7 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 8-Noded Solid Elements.

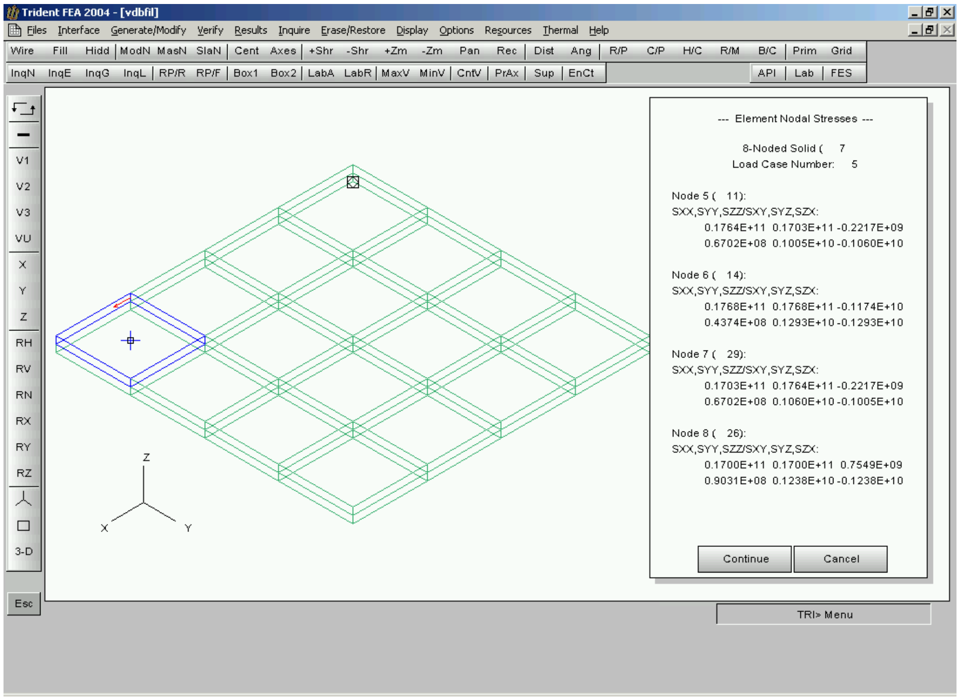


Figure 3.39: Nodal Stresses on Lower Surface of Element #7 for Geometrically Nonlinear Analysis of a Simply Supported Plate Using 8-Noded Solid Elements.

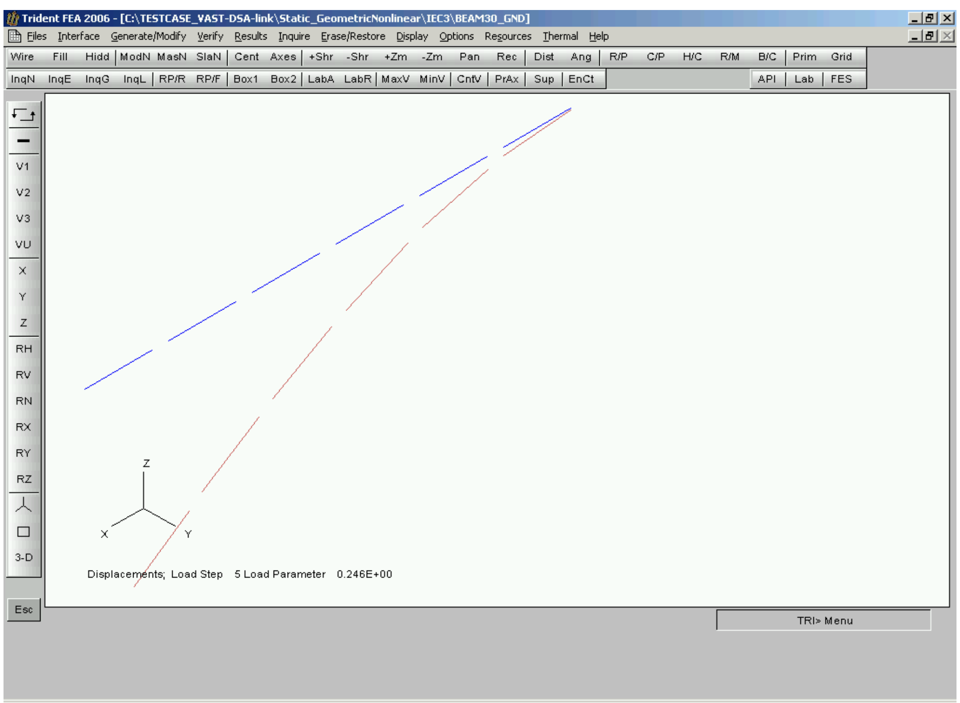


Figure 3.40: Original and Deformed Meshes for Geometrically Nonlinear Analysis of a Cantilever Beam Using 2-Noded General Beam Elements.

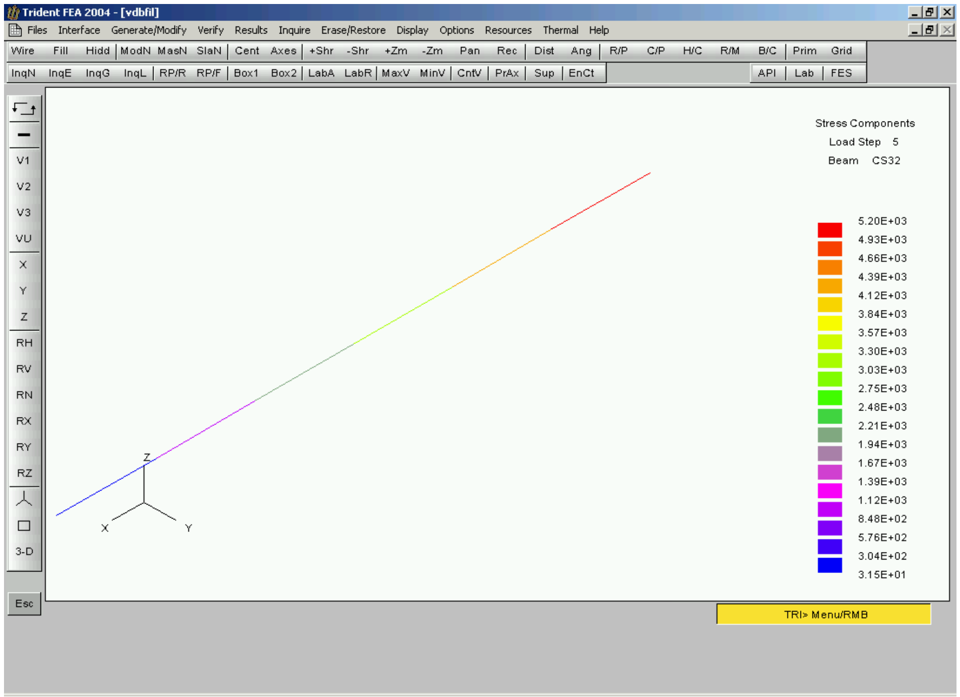


Figure 3.41: Element Stress Plot at Solution Step 5 for Geometrically Nonlinear Analysis of a Cantilever Beam Using 2-Noded Beam Elements.

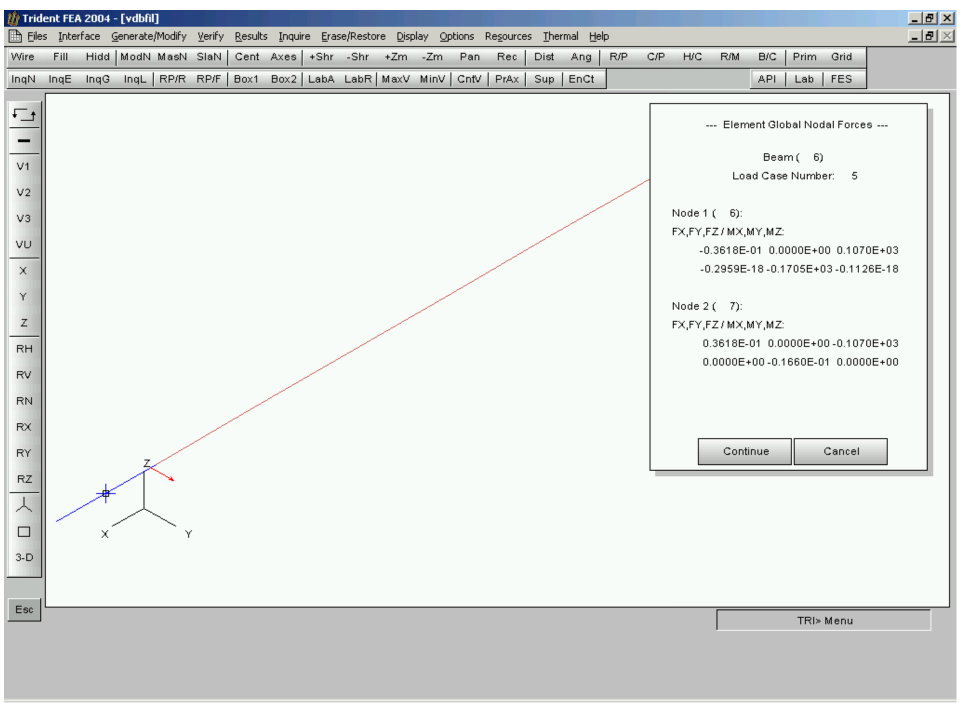


Figure 3.42: Nodal Forces in Element #6 at Solution Step 5 for Geometrically Nonlinear Analysis of a Cantilever Beam Using 2-Noded Beam Elements.

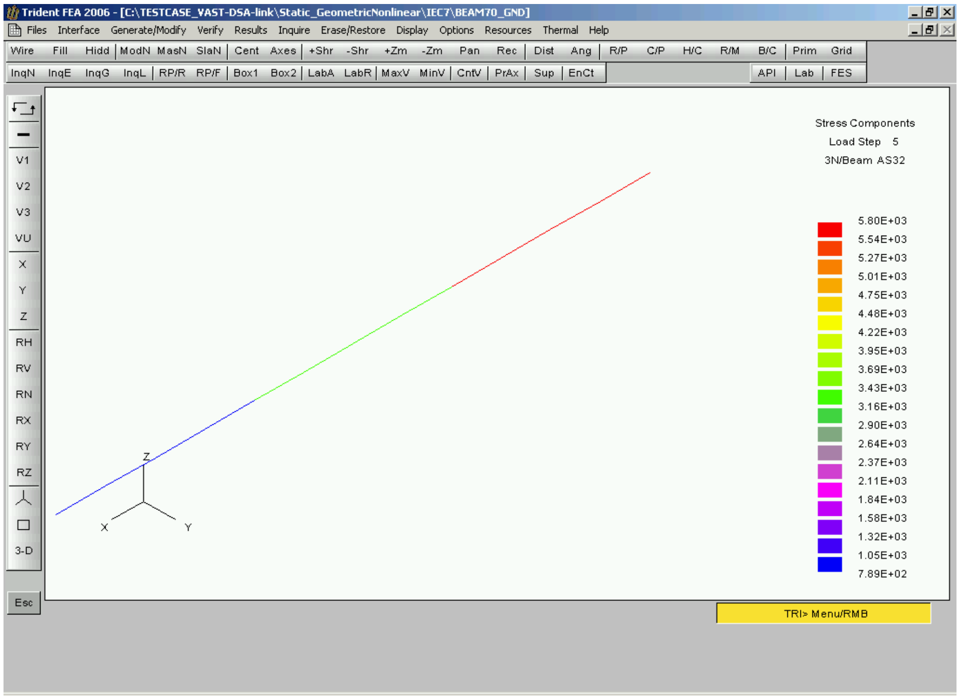


Figure 3.43: Element Stress Plot on Upper Surface for Geometrically Nonlinear Analysis of a Cantilever Beam Using 3-Noded Beam Elements.

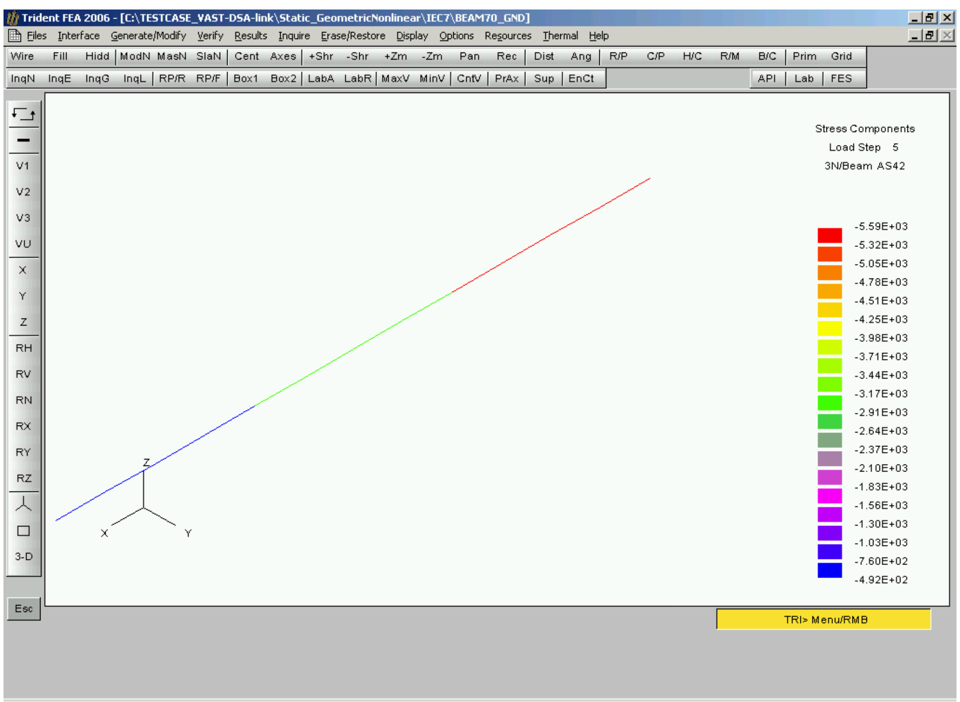


Figure 3.44: Element Stress Plot on Lower Surface for Geometrically Nonlinear Analysis of a Cantilever Beam Using 3-Noded Beam Elements.

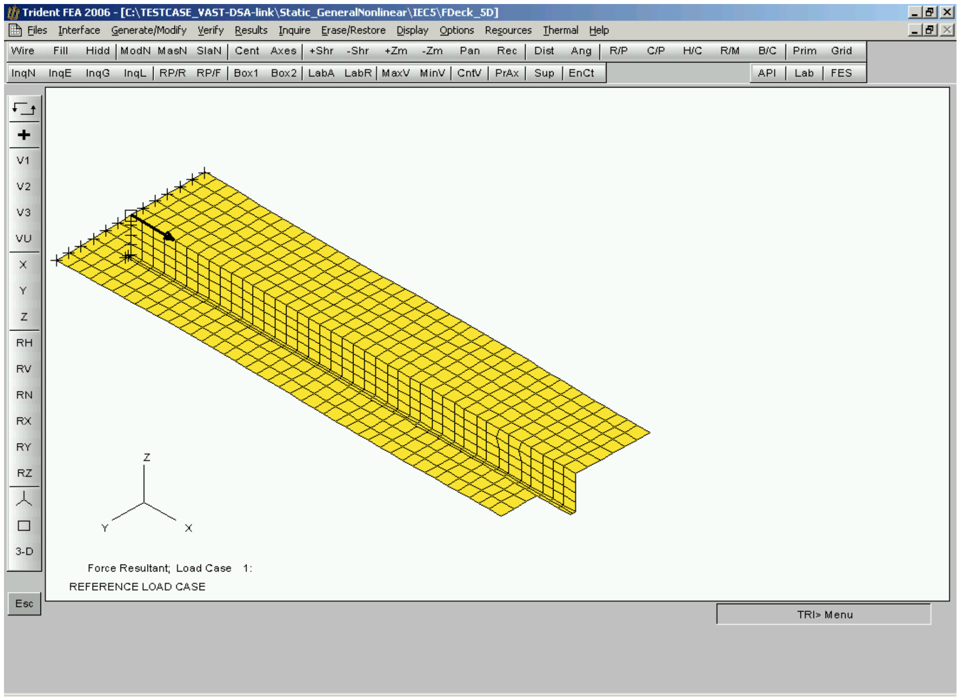


Figure 3.45: A 4-Noded Quad Shell Element Model for Elastic-Plastic Collapse Analysis of a Single Stiffened Panel.

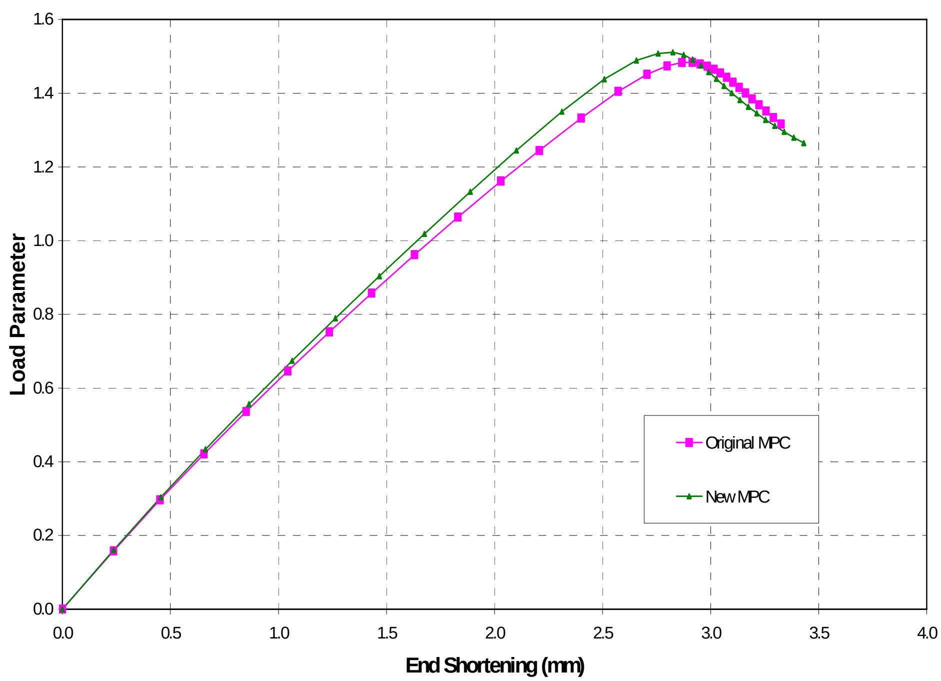


Figure 3.46: Load-Shortening Curves Obtained Using Original and Automatically Generated MPCs.

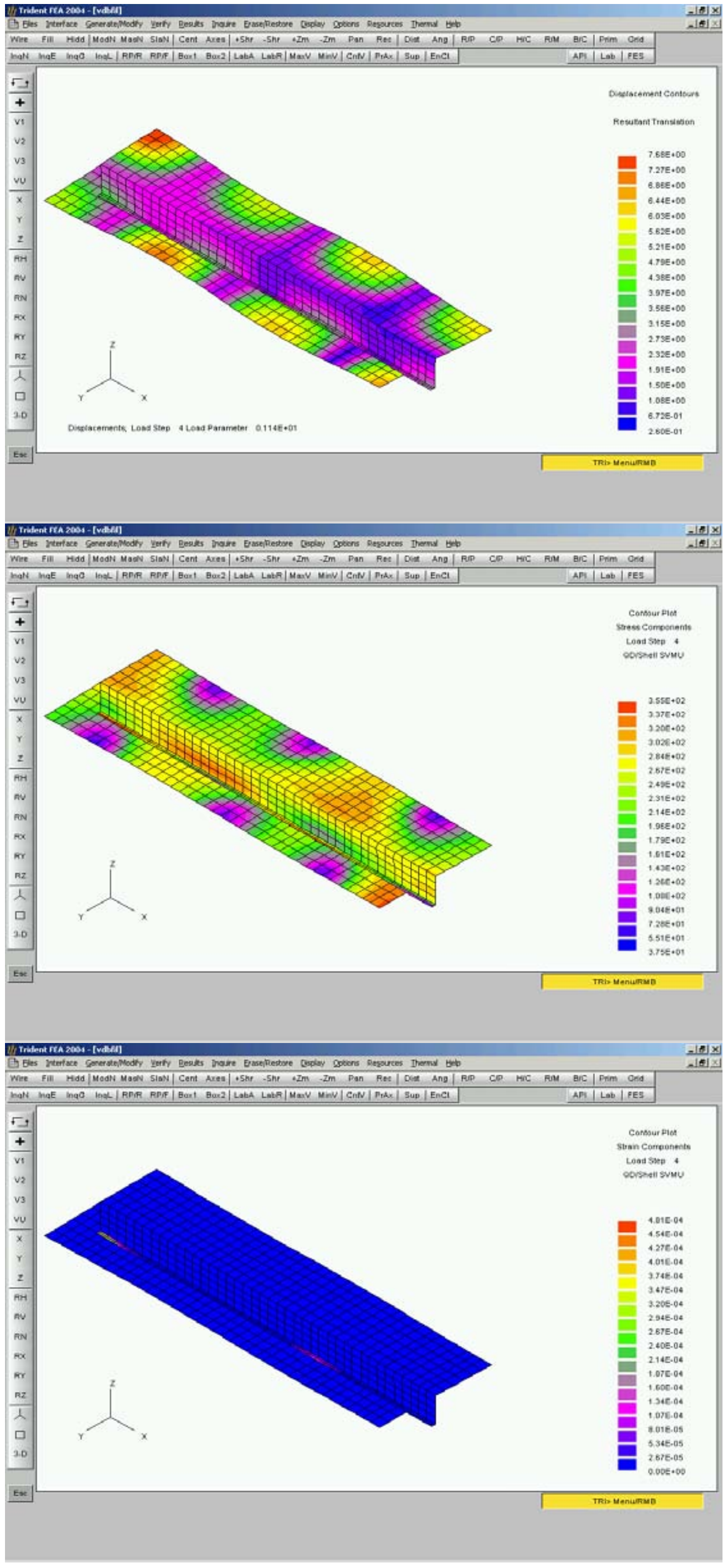


Figure 3.47: Nonlinear Solutions for the Stiffened Panel in the Pre-Buckling Range.

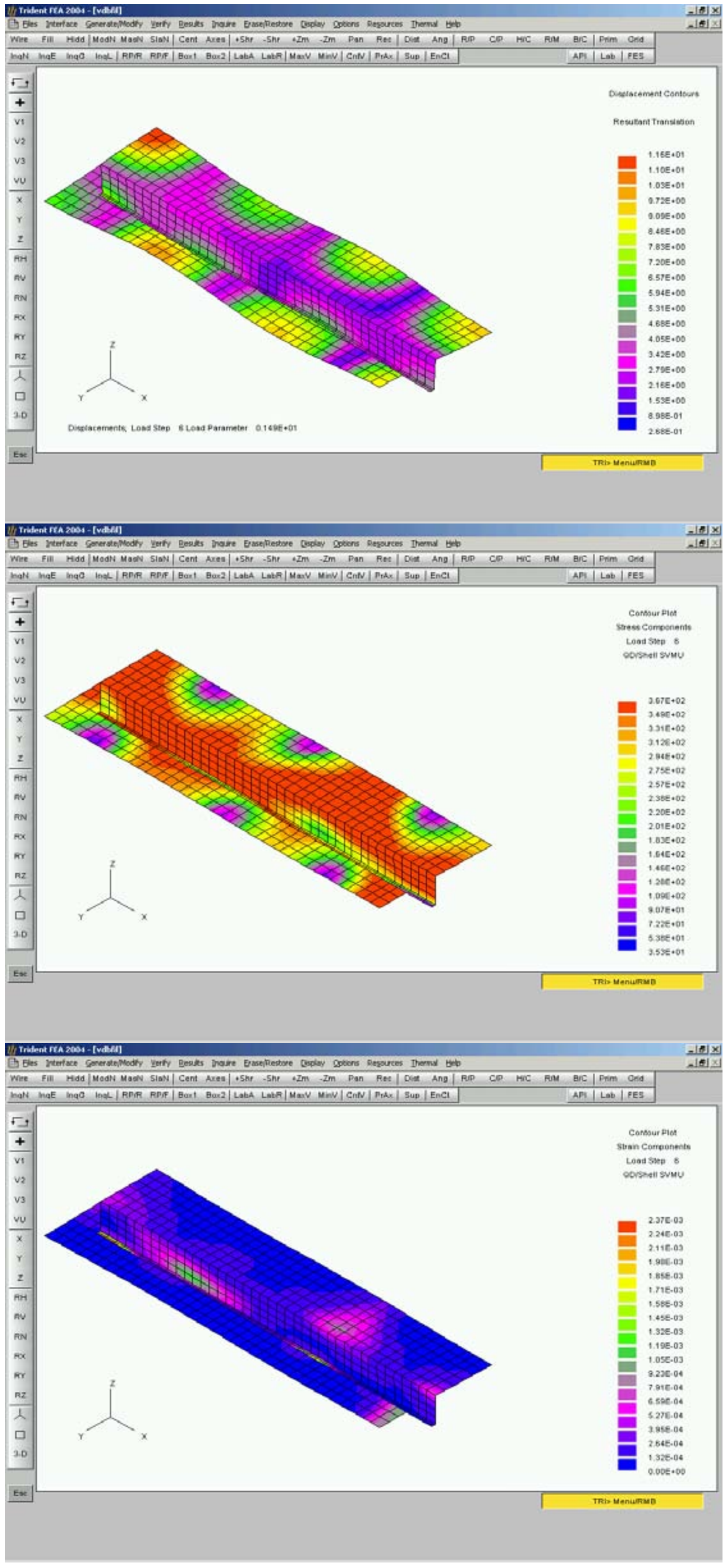


Figure 3.48: Nonlinear Solutions for the Stiffened Panel at the Limit Load Level.

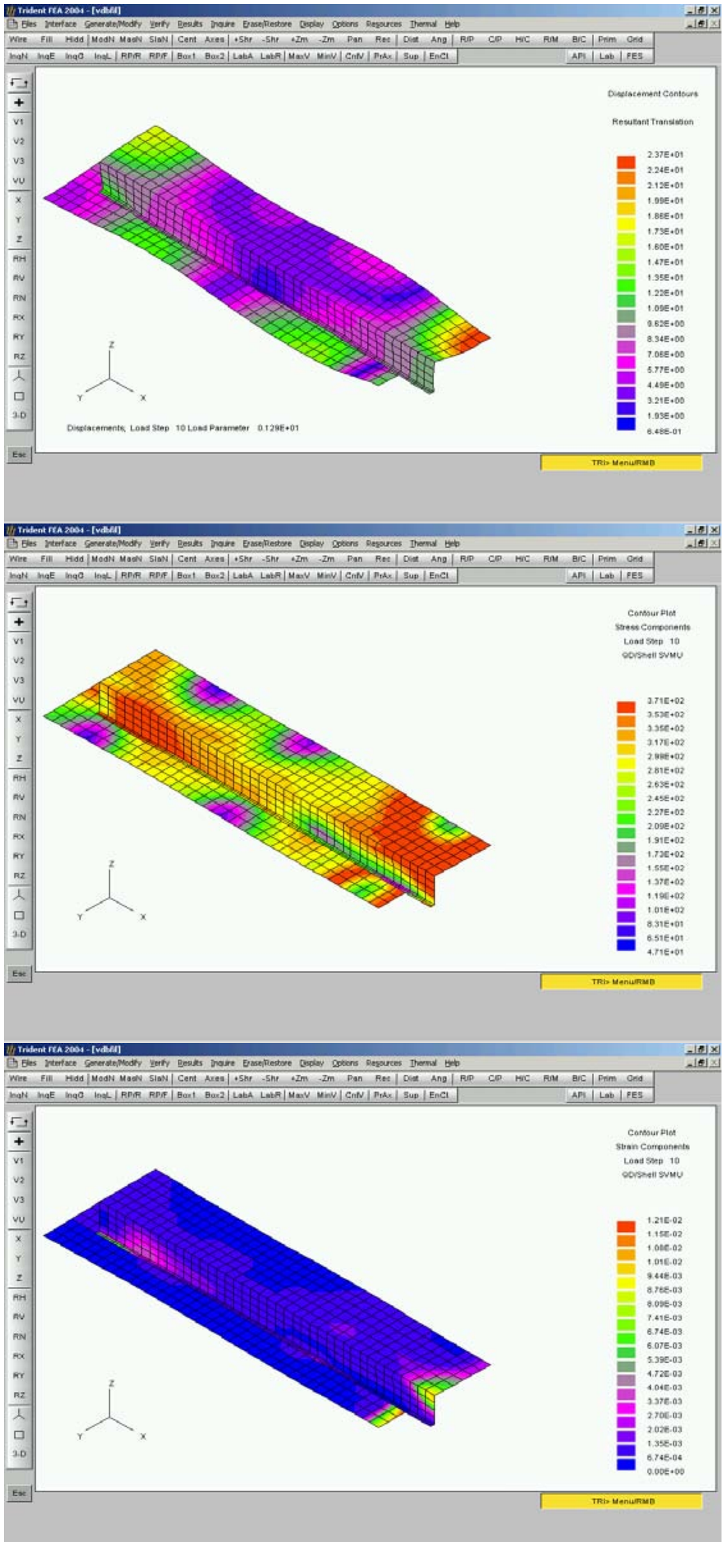


Figure 3.49: Nonlinear Solutions for the Stiffened Panel in Early Post-Buckling Range.

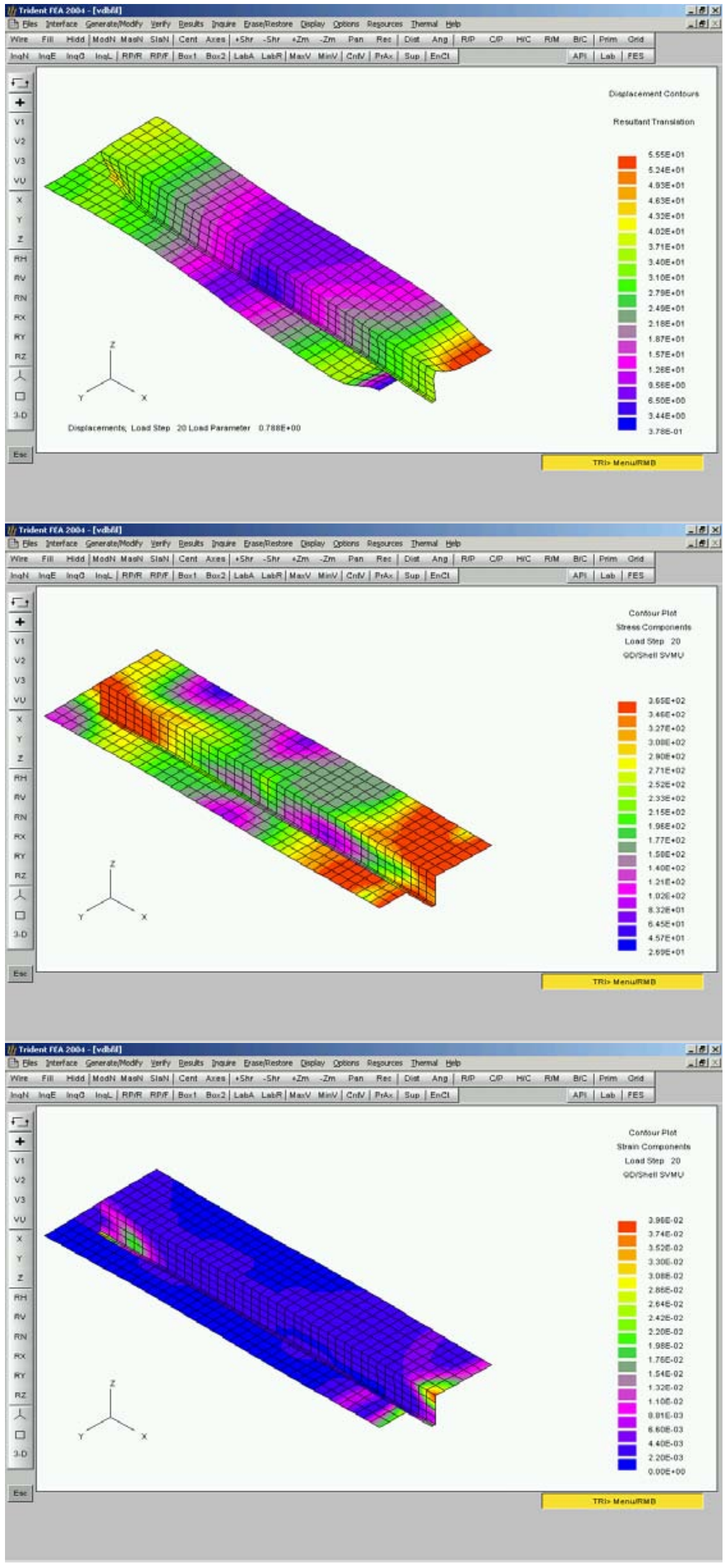


Figure 3.50: Nonlinear Solutions for the Stiffened Panel in Post-Buckling Range.

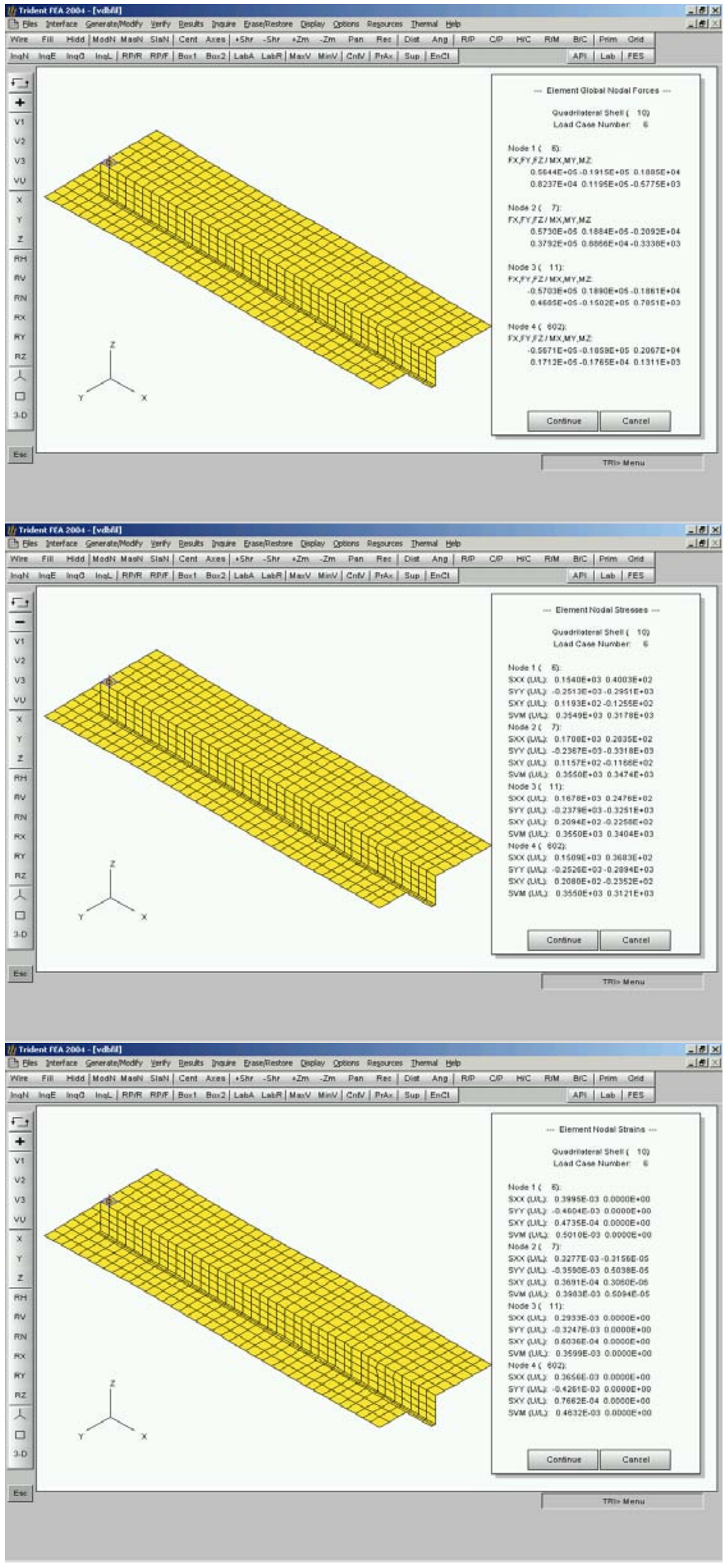


Figure 3.51: Nodal Forces, Nodal Stresses and Nodal Plastic Strains in Element #10.

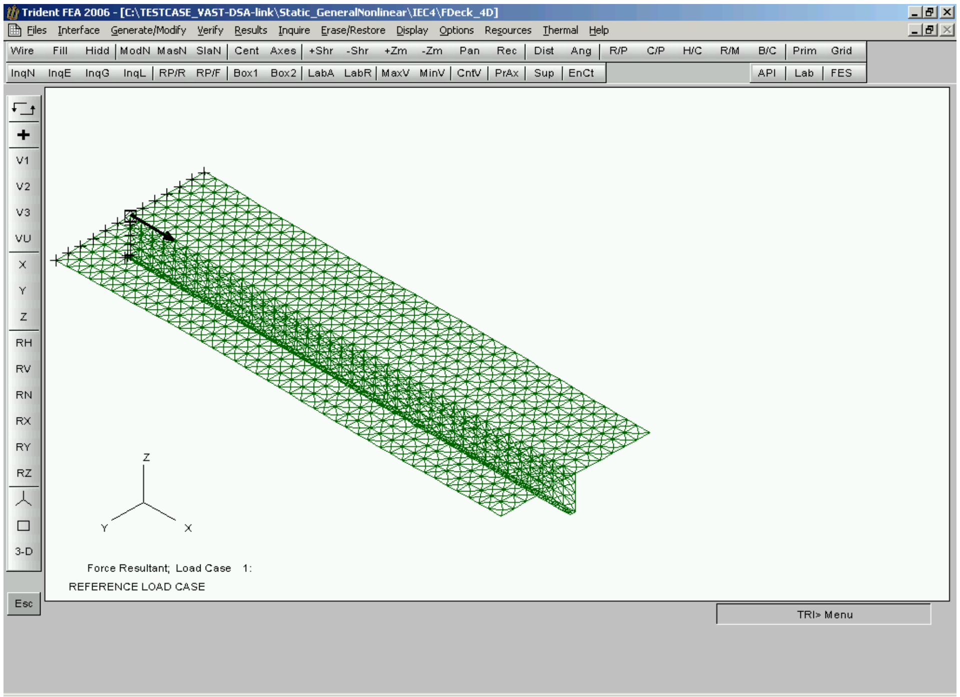


Figure 3.52: A 3-Noded Triangular Plate Element Model for Elastic-Plastic Collapse Analysis of a Single Stiffened Panel.

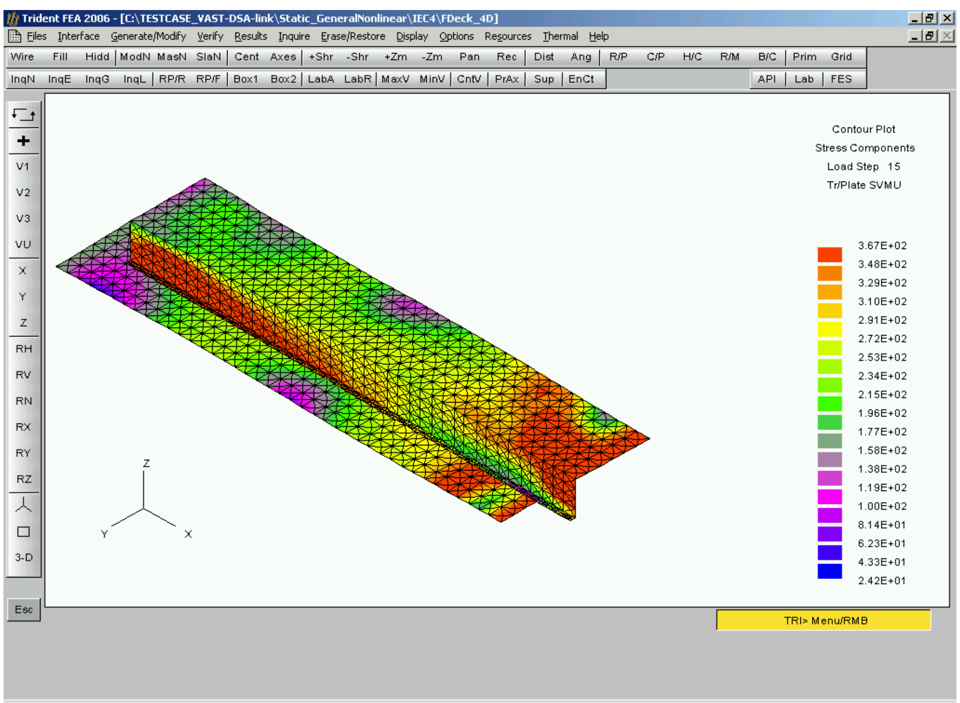


Figure 3.53: Stress Contour of the Single Stiffened Panel in Early Post-Buckling Range Predicted by the 3-Noded Triangular Plate Elements.

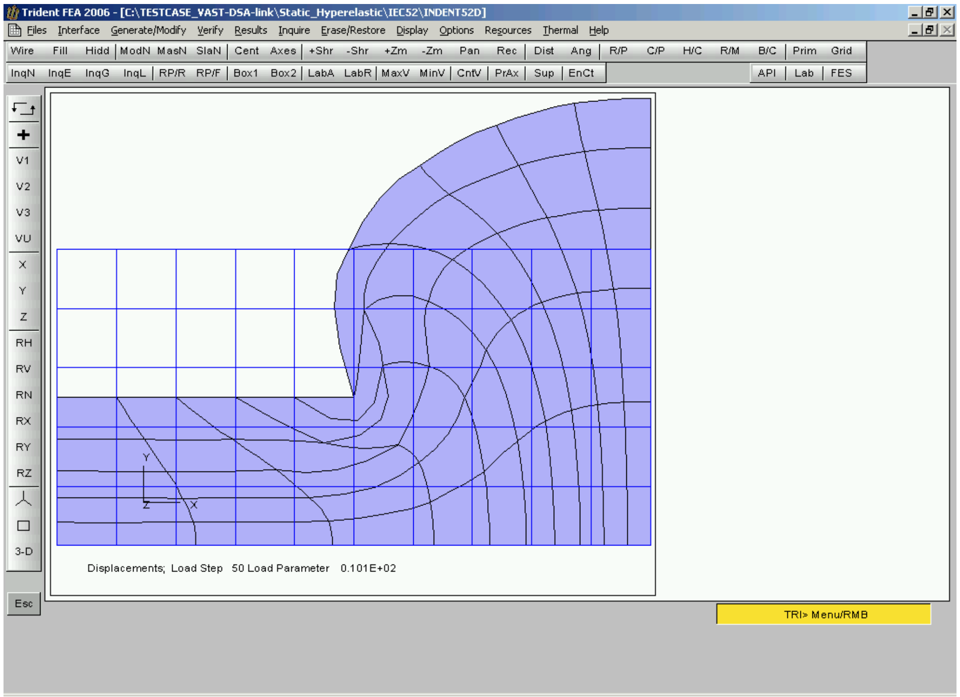


Figure 3.54: Original and Deformed Shapes of a Rubber Block Under Indentation by a Rigid Surface.

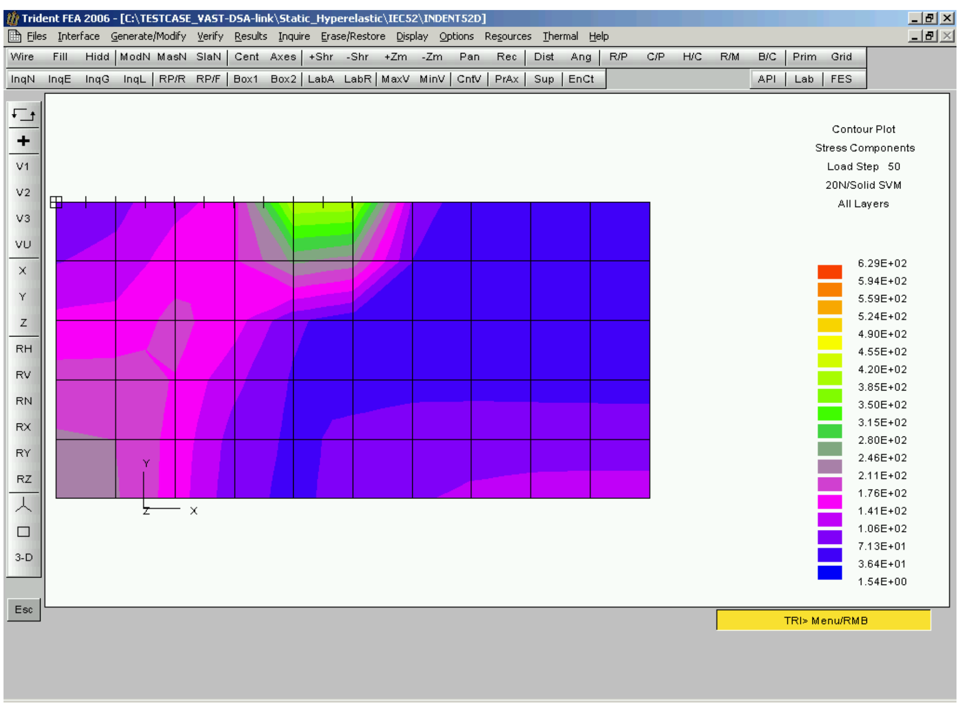


Figure 3.55: Stress Contour of the Rubber Block at Final Load Level.

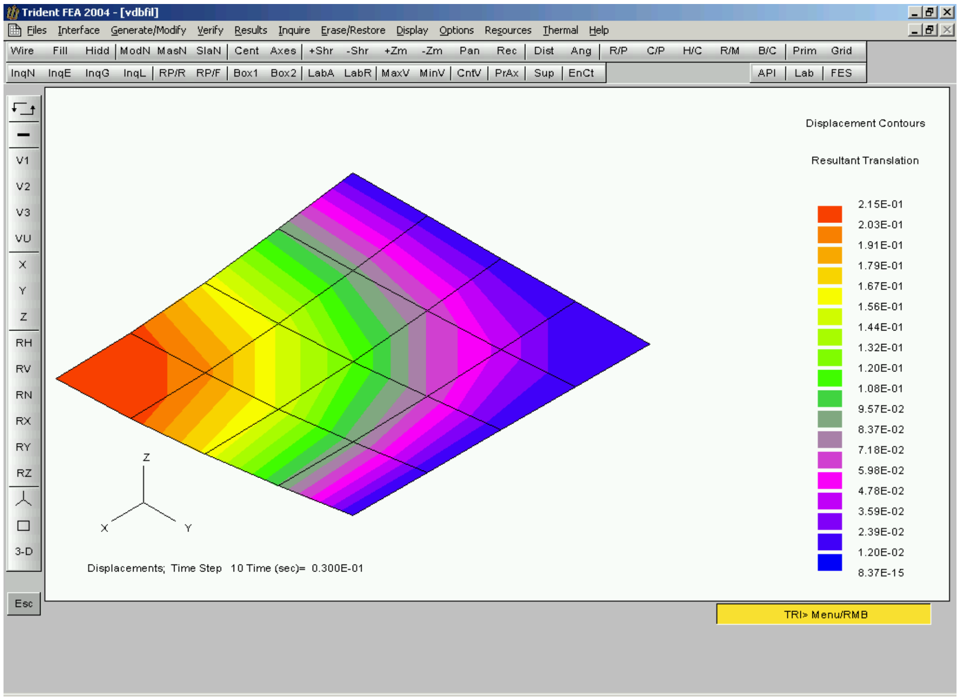


Figure 3.56: Deformed Shape and Displacement Contour at Time Step #10 of a Simply Supported Plate for Linear Dynamic Analysis.

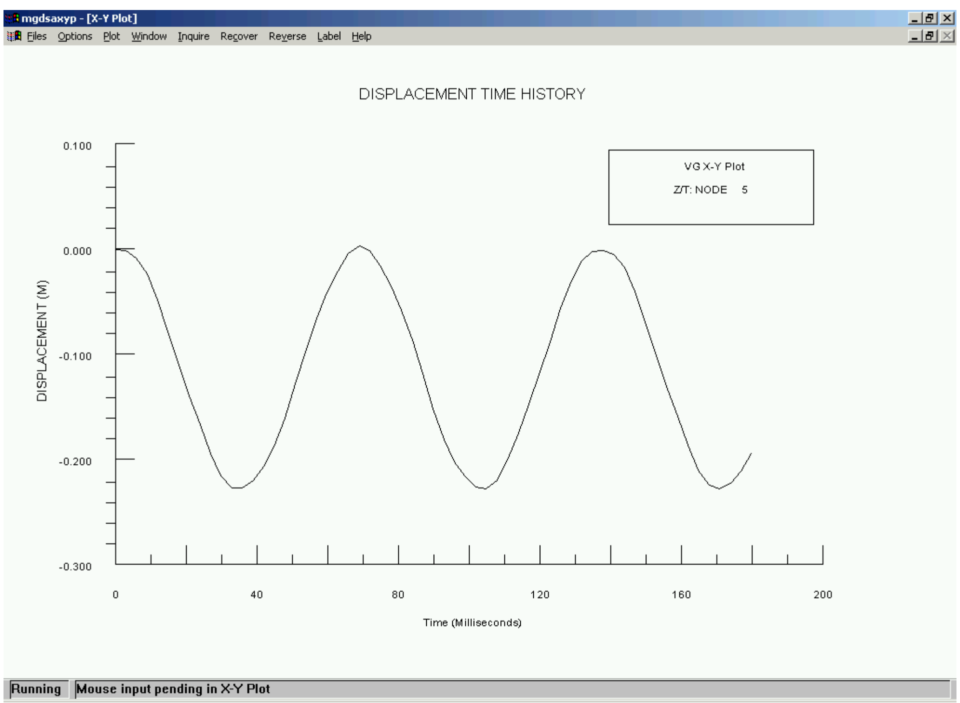


Figure 3.57: Time History of Center Deflection for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

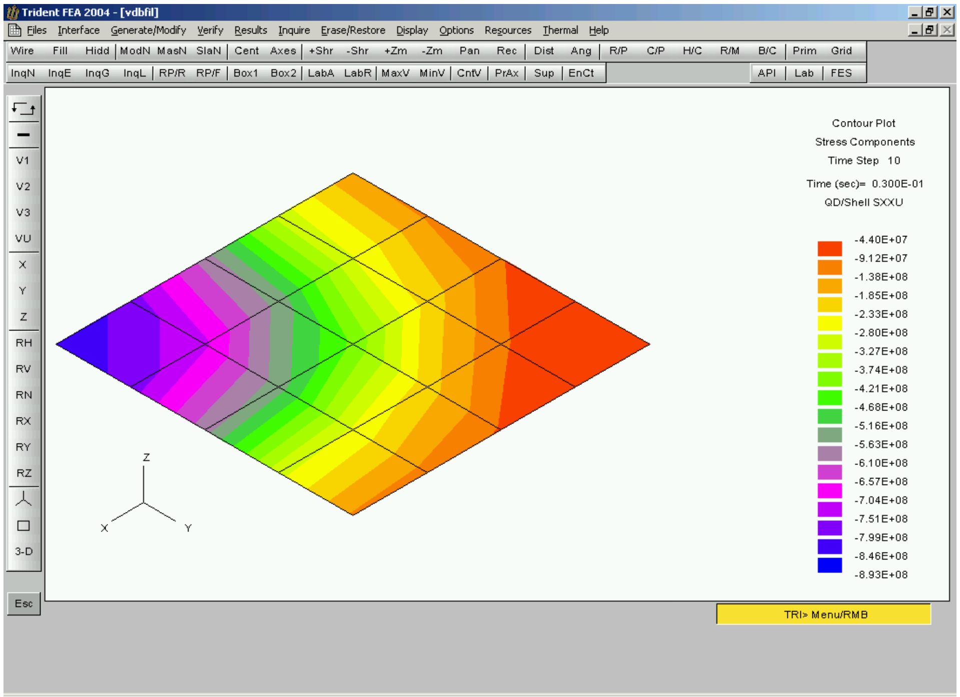


Figure 3.58: Stress Contour at Time Step #10 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

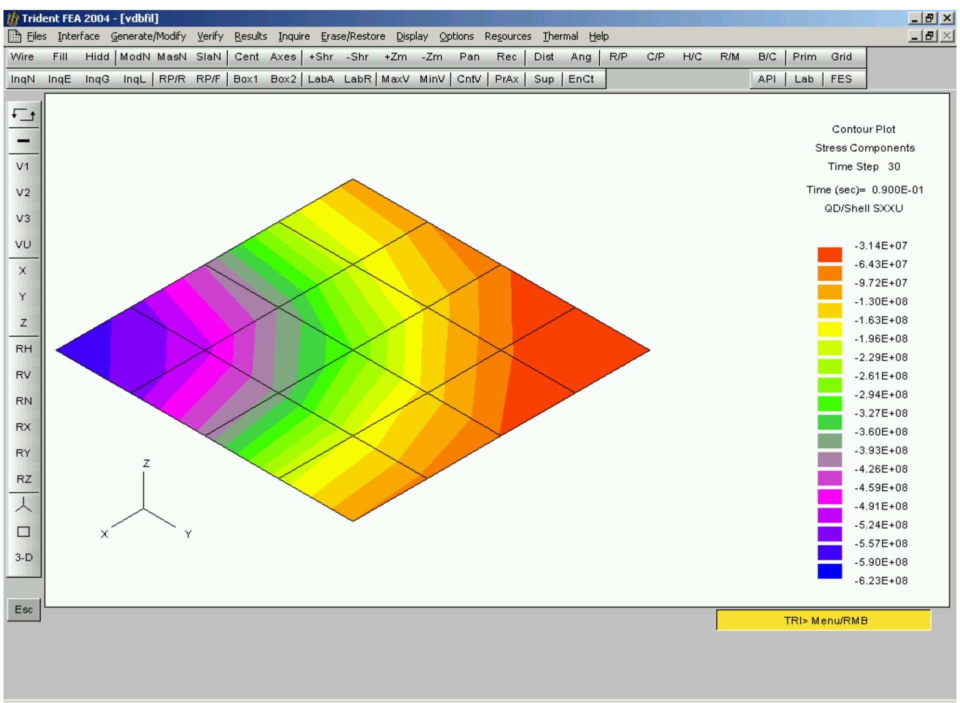


Figure 3.59: Stress Contour at Time Step #30 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

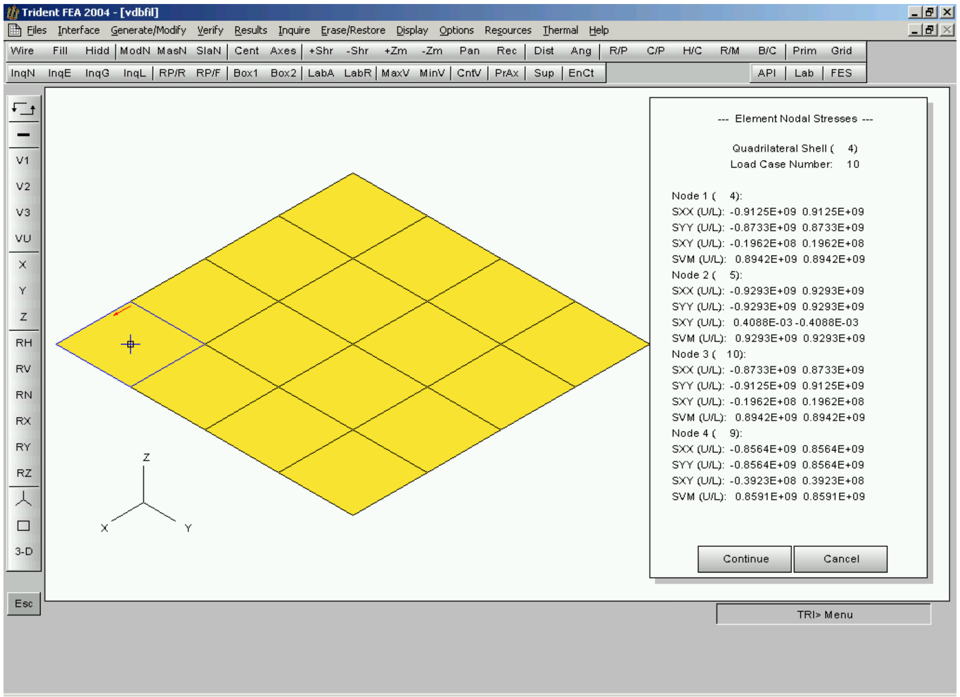


Figure 3.60: Nodal Stresses in Element #4 at Time Step 10 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

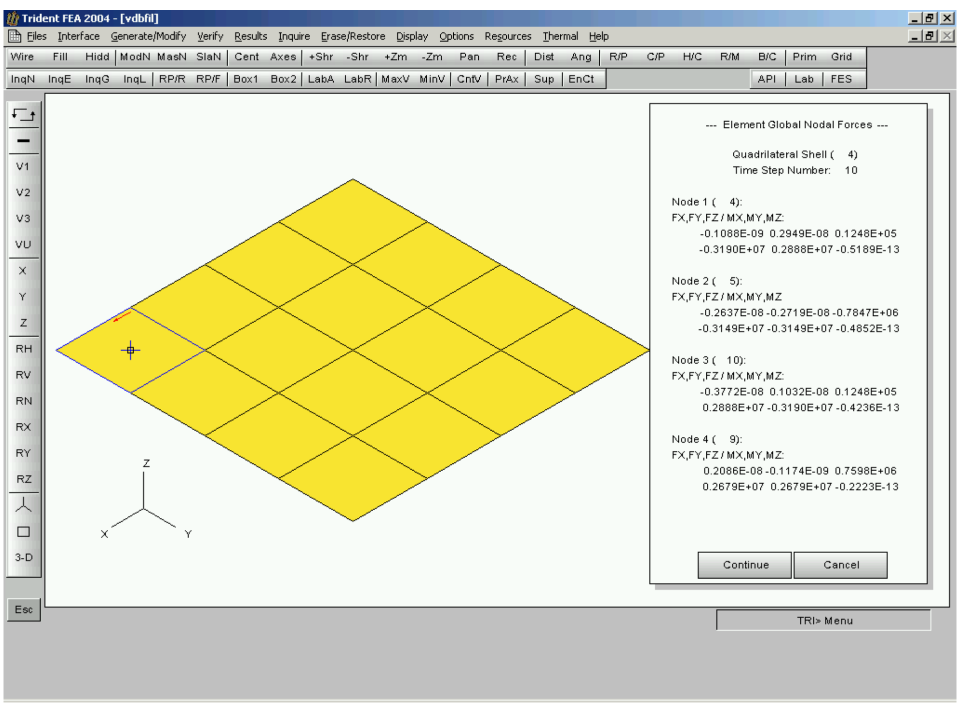


Figure 3.61: Nodal Forces in Element #4 at Time Step 10 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

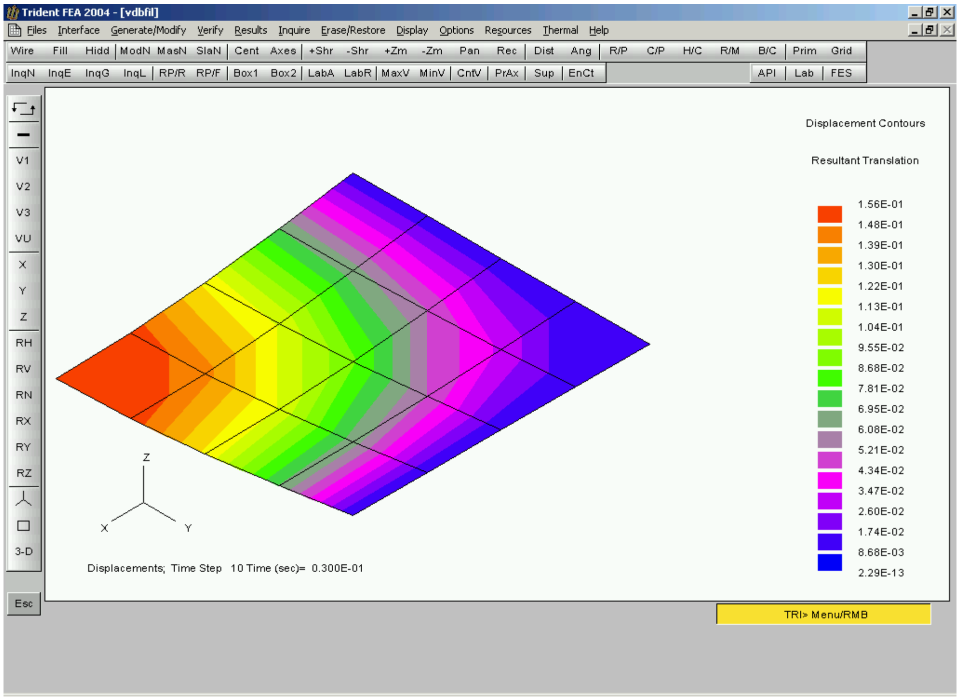


Figure 3.62: Deformed Shape and Displacement Contour at Time Step #10 of a Simply Supported Plate for Nonlinear Dynamic Analysis.

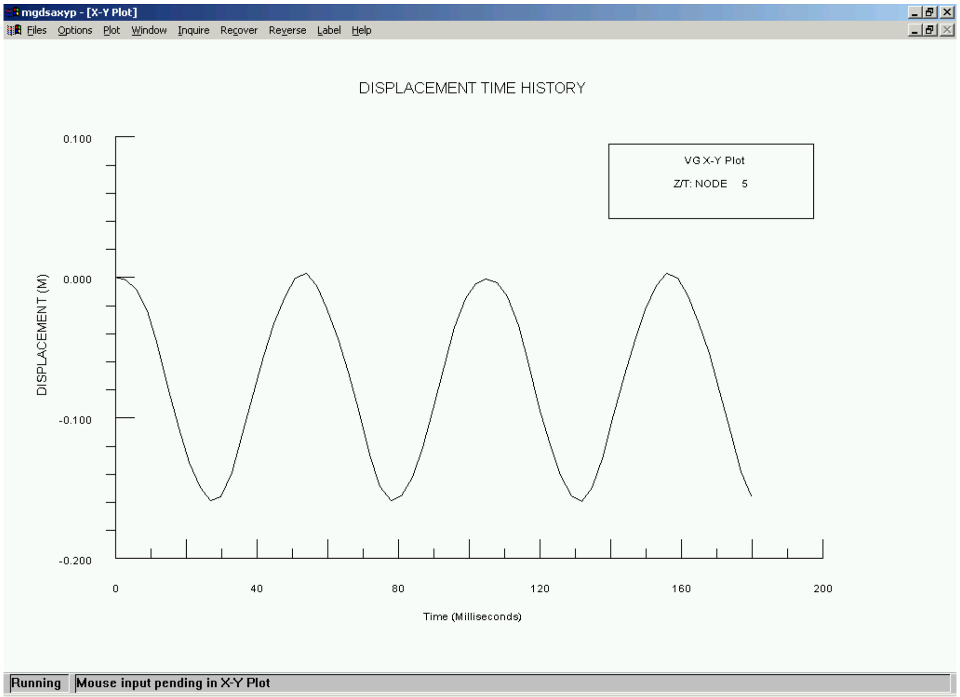


Figure 3.63: Time History of Center Deflection for Nonlinear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

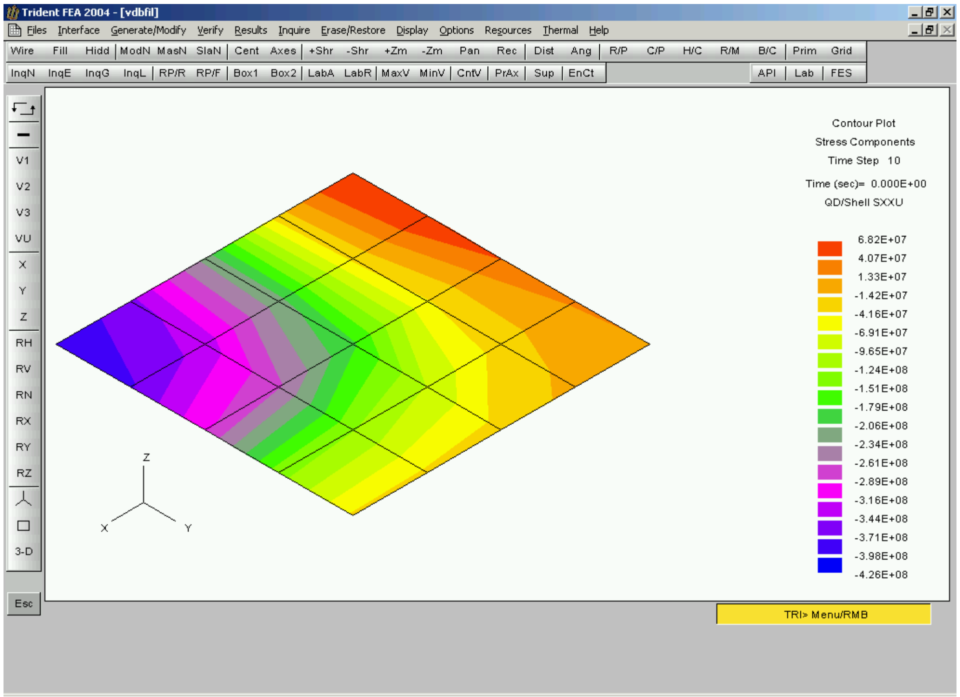


Figure 3.64: Stress Contour at Time Step #10 for Nonlinear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

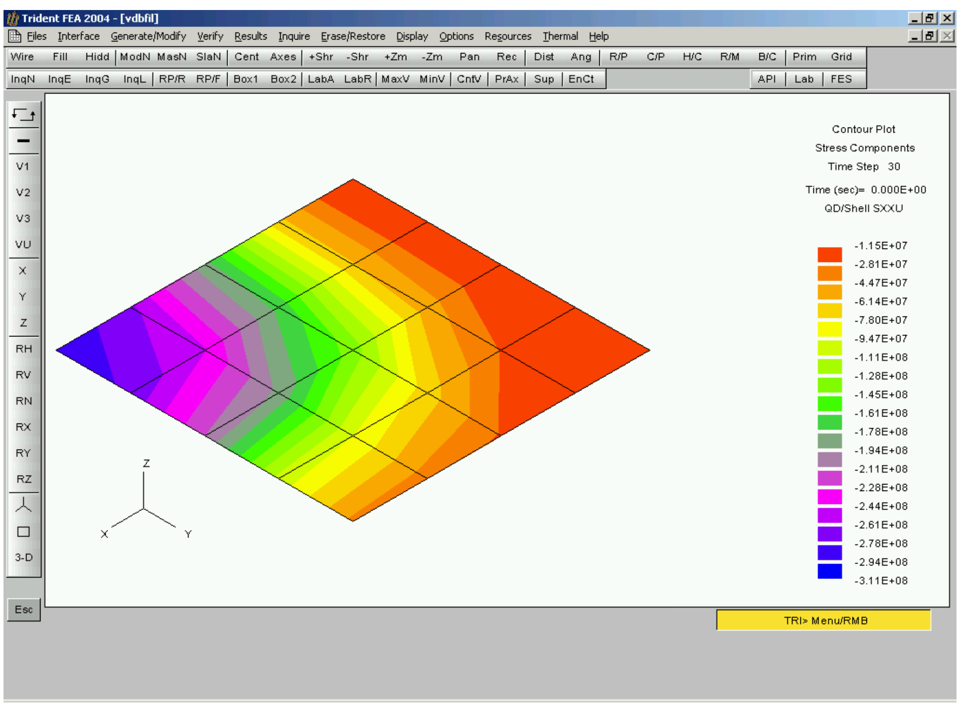


Figure 3.65: Stress Contour at Time Step #30 for Nonlinear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

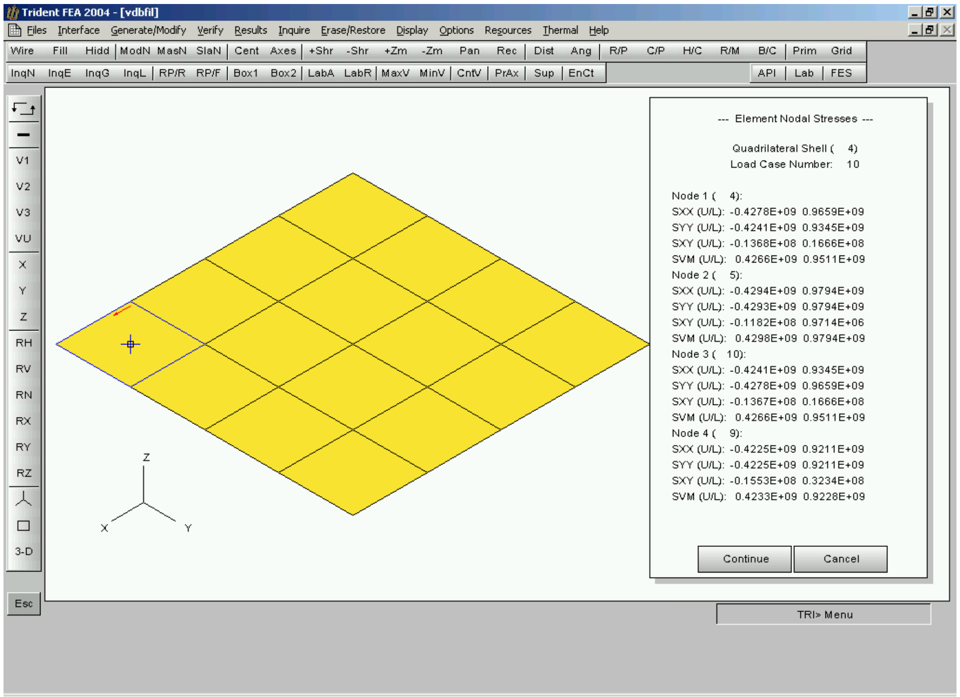


Figure 3.66: Nodal Stresses in Element #4 at Time Step 10 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

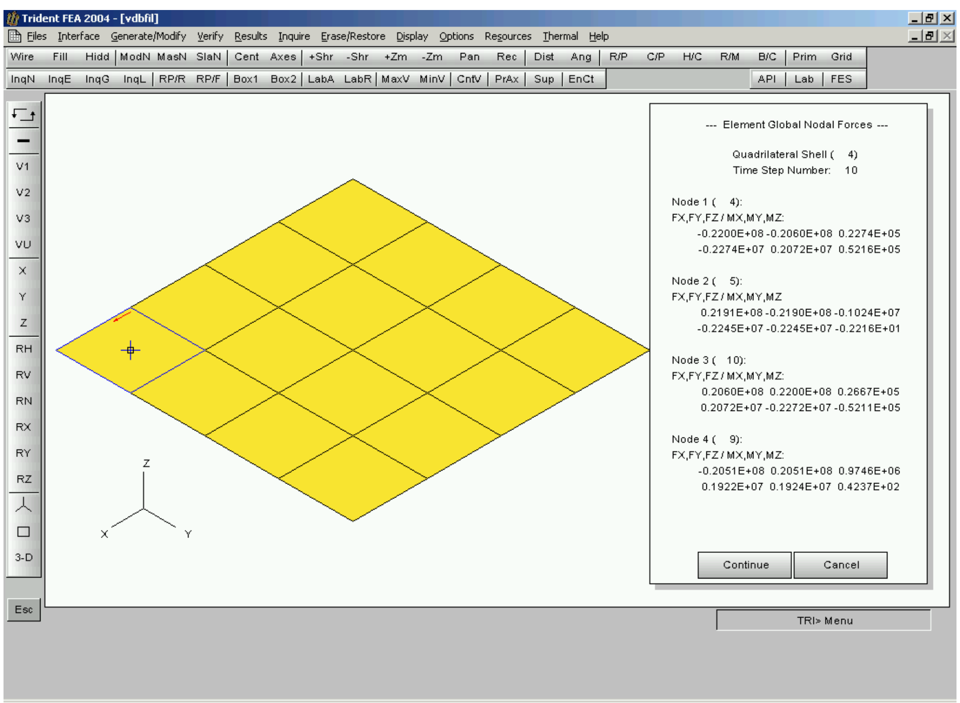


Figure 3.67: Nodal Forces in Element #4 at Time Step 10 for Linear Dynamic Response of a Simply Supported Plate Under a Sudden Application of Uniform Pressure.

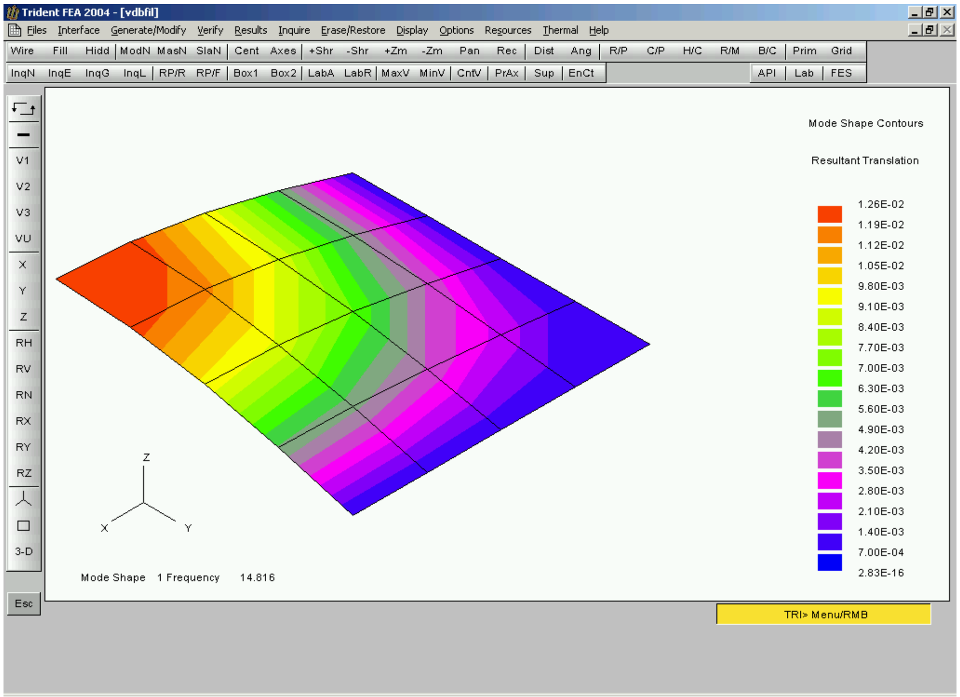


Figure 3.68: Mode Shape #1 of a Simply Supported Plate Modelled by 4-Noded Quad Shell Elements.

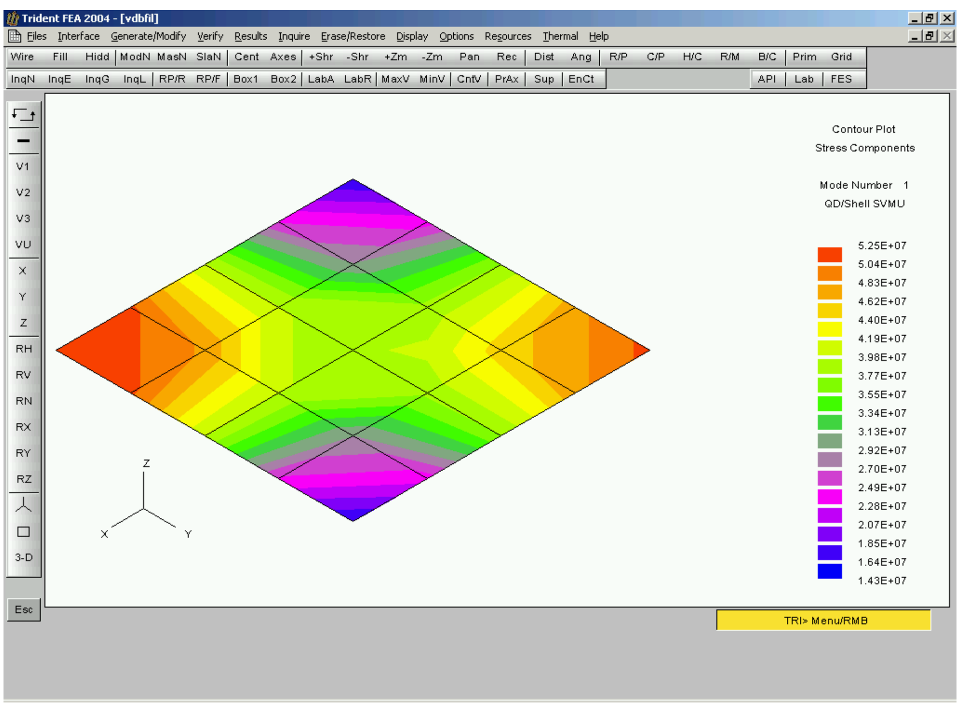


Figure 3.69: Modal Stress Contour for Mode #1 of a Simply Supported Plate.

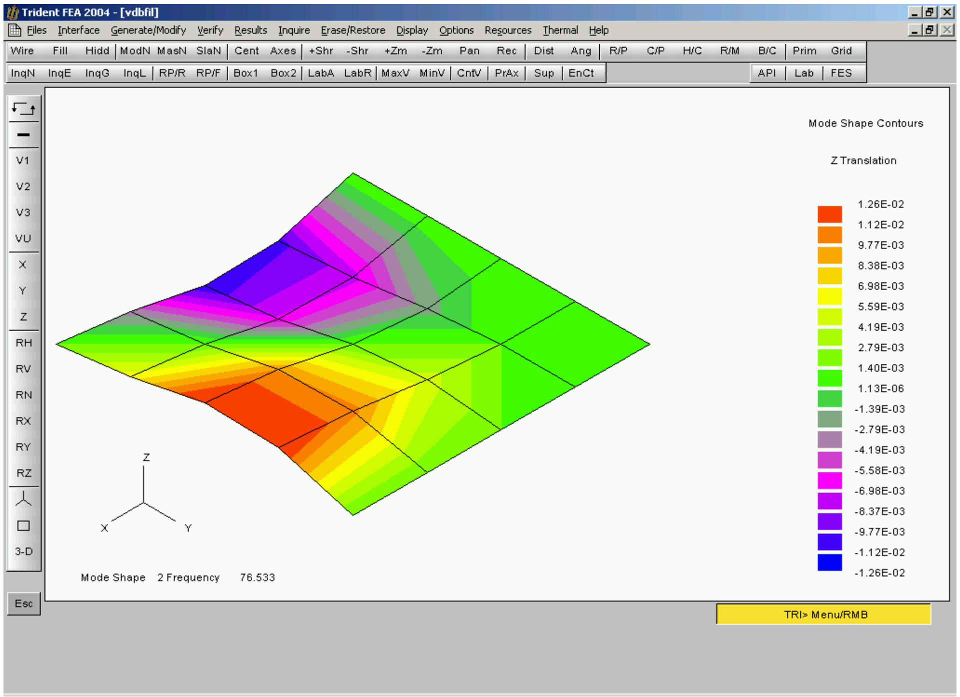


Figure 3.70: Mode Shape #2 of a Simply Supported Plate Modelled by 4-Noded Quad Shell Elements.

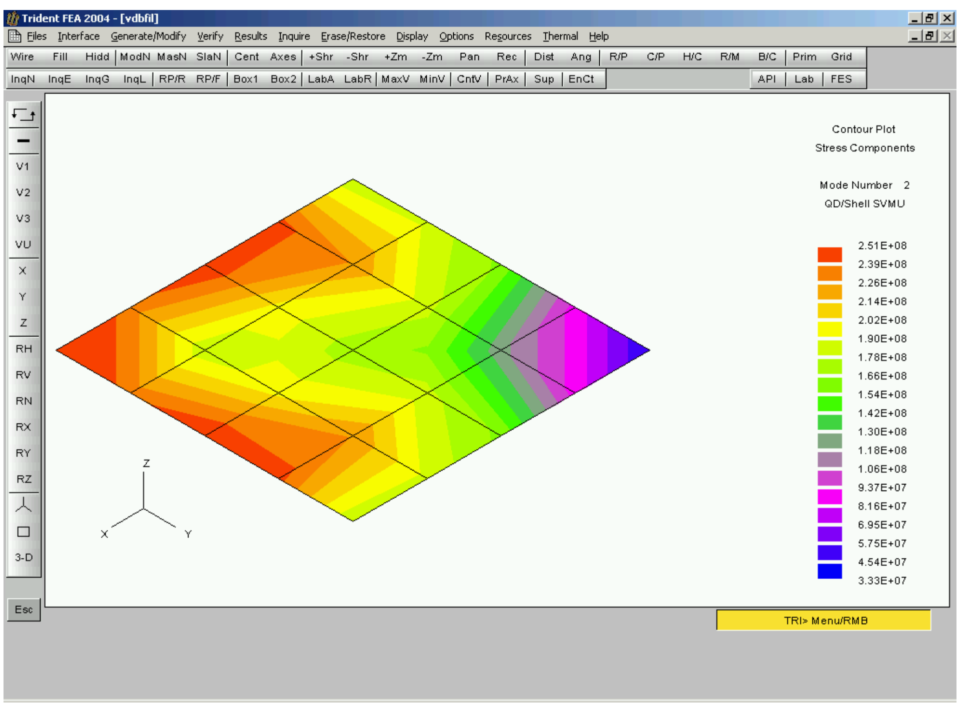


Figure 3.71: Modal Stress Contour for Mode #2 of a Simply Supported Plate.

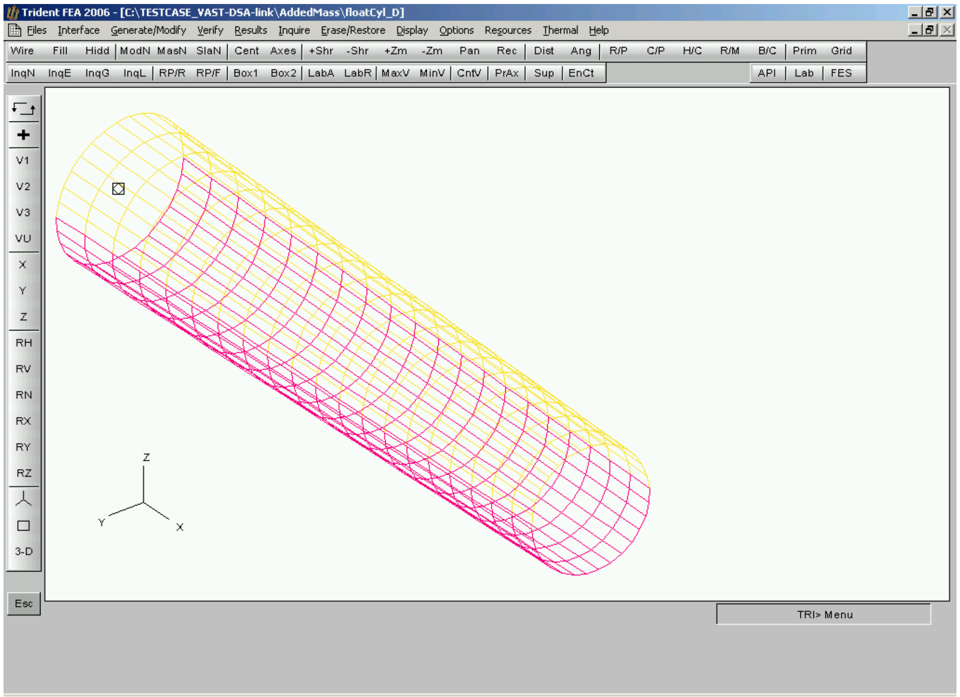


Figure 3.72: Structural and Fluid Element Meshes for Eigenvalue Analysis of a Floating Cylindrical Shell.

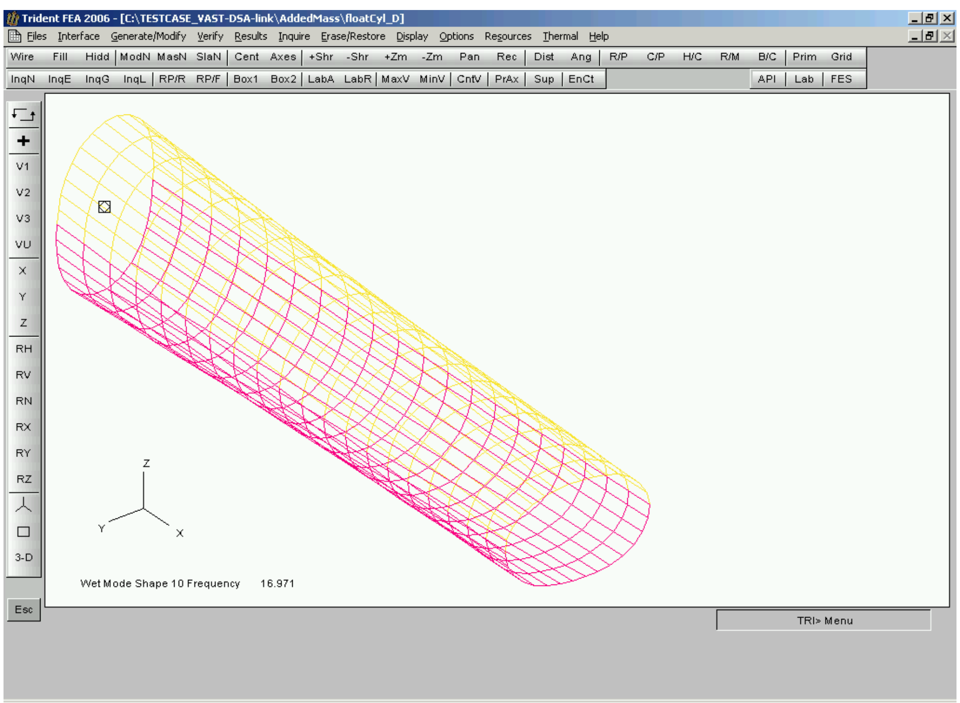


Figure 3.73: A Typical Natural Vibration Mode (#10) of the Floating Cylindrical Shell.

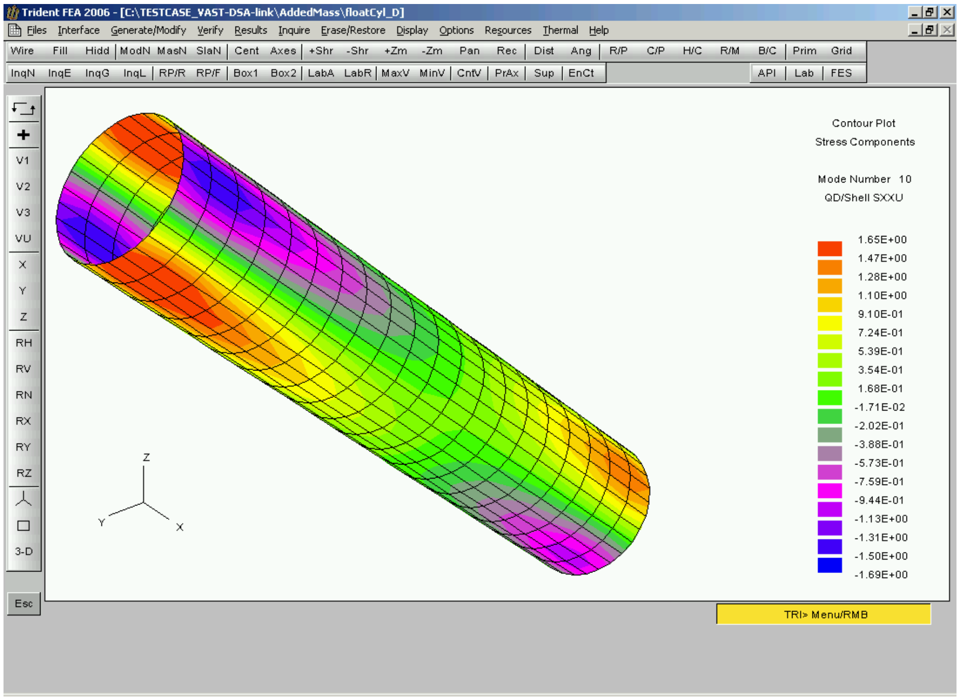


Figure 3.74: Modal Stress Contour for a Typical Mode Shape (#10) of the Floating Cylindrical Shell.

4. Conclusions

In this report, we have described development and verification of a direct linkage of the VAST program to the DSA database performed in a recent contract from DRDC Atlantic. This direct VAST-DSA link provides both pre- and post-processing capabilities to support finite element analyses using the VAST program. On one hand, it is capable of extracting input information required by VAST executions directly from the DSA database files, including nodal co-ordinates, element connectivity, concentrated and element load, lumped masses, fluid added masses and boundary conditions. On the other hand, it can automatically write computed element stresses and element internal forces to the appropriate stress files in DSA database. The most significant advantage of this direct VAST-DSA link is that it incorporates the most time-consuming parts of the pre- and post-processing operations into the VAST execution, which is normally carried out in batch mode.

This newly developed direct link between VAST and the DSA database was verified extensively using a large number of test cases covering all the commonly user element types and analysis options supported by the DSA database. The results are present in this report in detail, where the stress plots generated by using DSA stress files populated in different methods were compared and also verified against the stress results printed in the LPT file. The electronic files for all test cases are supplied along with this technical report.

One of the main objectives of the present contract was to speed up the process for generating the LOD file from the load information stored in the DSA database. To accomplish this, two new integer arrays were introduced in the DSA database routine for extracting load information to provide a mapping between non-zero concentrated and element load components to the nodes and elements on which these loads were applied. Use of these arrays reduced I/O operations significantly and increased speed for LOD file generation by a factor of 5 to 10.

A benchmark on the performance of the direct VAST-DSA link for post-processing was also performed using a global CPF model with 207 load cases. The benchmark results indicated that when multiple load cases existed, the I/O operations to the DSA database files, VES, VEF and VNS, were less efficient than that to the VAST stress file, V53. However, use of the direct VAST-DSA link still has the benefit of having the stress transfer done as a part of the VAST run. To further improve numerical efficiency for populating the DSA stress files, the format of the VES, VEF and VNS may need to be modified.

Other tasks accomplished in this contract included (1) update of the Ansys/DSA translator and (2) enhancement of the HOODFEM program to provide subroutines for extracting mass values and center of gravity co-ordinates for each finite element from the DSA database.

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) MARTEC Limited, 1888 Brunswick Street, Suite 400, Halifax, Nova Scotia, Canada B3J 3J8	2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable). UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Direct Linkage of the VAST Finite Element Code to the DSA Database

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Jiang L., Norwood M.

5. DATE OF PUBLICATION (month and year of publication of document) September 2006	6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc). 83	6b. NO. OF REFS (total cited in document) 0

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

CONTRACT REPORT

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R&D Canada - Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant). 11 gk 14, 11 gt 4	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written). W7707-05-3119/001/HAL
10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) Martec Technical Report TR-06-30	10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.) DRDC Atlantic CR 2006-185

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

UNLIMITED

13. ABSTRACT

The general-purpose finite element code VAST, which was developed and maintained by Martec Limited over the past three decades under sponsorship from DRDC Atlantic, has been adopted as the finite element solver engine in several ship structural analysis programs, including the STRUC code developed at DRDC Atlantic. The execution of the VAST program required a set of ASCII input data files and the calculated element stresses were stored in a binary file, named Prefix.V53. In order to assist model generation and stress post-processing, a GUI system, named Trident FEA, has been developed by Martec Limited based on the DSA database. Because a direct linkage between the VAST program and the DSA database was not yet established, all the VAST input data files must be created by Trident or STRUC using information stored in the DSA database and the stress results generated by VAST must be transferred back to the DSA database for plotting. These I/O-intensive, time-consuming operations had to be carried out in an interactive mode, causing a serious efficiency problem for large finite element models with many load cases.

In this contract, a direct linkage of the VAST program to the DSA database was developed. This direct VAST-DSA link provided not only a pre-processing capability for extracting all input information required for VAST executions directly from the DSA database files, but also a post-processing capability to automatically write computed element stresses and element internal forces to the appropriate stress files in DSA database. The most significant advantage of this direct VAST-DSA link was that it integrated the most time-consuming parts of the pre- and post-processing operations into the VAST execution, which was normally conducted in batch mode. This direct link between VAST and the DSA database was verified extensively using a large number of test cases covering all the commonly used element types and analysis options supported by the DSA database. The results are present in this report in detail.

One of the main objectives of the present contract was to speed up the process for generating the LOD file from the load information stored in the DSA database. This was accomplished through modifications in the DSA database routine for extracting load information. Numerical examples indicated that these modifications resulted in a significant reduction on I/O operations, so the speed for LOD file generation was increased by a factor of 5 to 10.

A benchmark on the performance of the direct VAST-DSA link for post-processing was also performed using a practical finite element model. The benchmark results indicated that when multiple load cases existed, the I/O operations on the DSA database files, VES, VEF and VNS, were less efficient than that on the VAST stress file, V53. However, use of the direct VAST-DSA link still has the benefit of having the stress transfer done as a part of the VAST run. To further improve numerical efficiency for populating the DSA stress files, the format of the VES, VEF and VNS may need to be modified.

Other tasks accomplished in this contract included (1) update of the Ansys/DSA translator and (2) enhancement of the HOODFEM program to provide subroutines for extracting mass values and center of gravity co-ordinates for each finite element from the DSA database.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS

finite element analysis
ship structural response
structural stress
DSA database
VAST finite element code

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca